



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

DATALOGGER S ROZHRANÍM ETHERNET

DATALOGGER WITH ETHERNET

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Tomáš Orávik

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Zdeněk Bradáč, Ph.D.

BRNO 2016

Diplomová práce

magisterský navazující studijní obor **Kybernetika, automatizace a měření**

Ústav automatizace a měřicí techniky

Student: Bc. Tomáš Orávik

ID: 136568

Ročník: 2

Akademický rok: 2015/16

NÁZEV TÉMATU:

Datalogger s rozhraním Ethernet

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte a realizujte elektronický datalogger vybavený rozhraním Ethernet. Datalogger založte na mikrokontrolerové platformě Raspberry PI. Datalogger vybavte digitálními a analogovými V/V, záznamovým médii na bázi FLASH karty a komunikačním rozhraním. Navrhněte a realizujte HW periferních obvodů, navrhněte a realizujte DPS, oživte a otestujte. Vytvořte komplexní programové vybavení pro mikrokontrolér a pro nadřazené PC. Realizujte i přístup do dataloggeru přes WWW rozhraní. Demonstrujte správnou funkci.

1. Proveďte literární rešerši.
2. Navrhněte a realizujte koncepci systému.
3. Vytvořte vhodné rozhraní a připojení pro analogové a digitální sensory a akční členy a dále implementujte vhodné úložné médium.
4. Navrhněte a realizujte elektroniku systému, realizujte DPS, osadte je součástkami a oživte.
5. Vytvořte komplexní programové vybavení, které umožní ovládání a administrování systému.
6. Otestujte funkčnost a demonstруйте ji, sepište diplomovou práci.

DOPORUČENÁ LITERATURA:

Pavel Herout: Učebnice jazyka C, KOPP, 2004, IV. přepracované vydání, ISBN 80-7232-220-6

Dle pokynů vedoucího práce.

Termín zadání: 8.2.2016

Termín odevzdání: 16.5.2016

Vedoucí práce: doc. Ing. Zdeněk Bradáč, Ph.D.

Konzultant diplomové práce:

doc. Ing. Václav Jirsík, CSc., předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Práca sa zaoberá návrhom dataloggeru s ethernetovým rozhraním riadeného pomocou Raspberry Pi 2. Datalogger je vybavený ôsmimi digitálnymi vstupmi a tromi analógovými vstupmi. Umožňuje ukladanie nameraných dát do databázy na microSD kartu. Používa komunikáciu po ethernetovom rozhraní v Raspberry Pi. Administrácia a ovládanie je umožnené cez webový server.

Kľúčové slová

Datalogger, microSD karta, digitálne vstupy, analógové vstupy, AD prevodník, Raspberry Pi 2, web server, SPI, I²C, MySQL, PHP, HTML, JavaScript, Python,

Abstrakt

Thesis deals with Datalogger with Ethernet interface controlled by Raspberry Pi 2. Datalogger is equipped with eight digital inputs and three analogue inputs. It allows storage of measured data into the database on the microSD card. It communicates over an Ethernet interface on the Raspberry Pi. The administration and control of the datalogger is possible through a web server.

Keywords

Datalogger, microSD card, digital inputs, analog inputs, AD converter, Raspberry Pi 2, web server, SPI, I²C, MySQL, PHP, HTML, JavaScript, Python

Bibliografická citácia:

ORÁVIK, T. *Datalogger s rozhraním Ethernet*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2016. 75 s. Vedúci diplomovej bol práce doc. Ing. Zdeněk Bradáč, Ph.D.

Prehlásenie

„Prehlasujem, že svoju diplomovú prácu na tému Datalogger s rozhraním Ethernet som vypracoval samostatne pod vedením vedúceho diplomovej práce a s použitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej diplomovej práce ďalej prehlasujem, že v súvislosti s vytvorením tejto diplomovej práce som neporušil autorské práva tretích osôb, hlavne som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a som si plne vedomý následkov porušenia ustanovení § 11 a nasledujúcich autorského zákona č. 121/2000 Zb., vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovení časti druhej, hlavy VI., diel 4 Trestného zákonníka č. 40/2009 Zb.

V Brne, dňa 16.5.2016

.....
podpis autora

Pod'akovanie

Ďakujem vedúcemu diplomovej práce doc. Ing. Zdeňek Bradáč, Ph.D za účinnú metodickú, pedagogickú a odbornú pomoc a ďalšie cenné rady pri spracovaní mojej diplomovej práce.

V Brne dňa: 16.5.2016

.....
podpis autora

Obsah

1	Úvod	11
2	Datalogger	12
2.1	Rozdelenie dataloggerov	12
2.1.1	Jednoúčelový datalogger	12
2.1.2	Viacúčelový datalogger	12
2.1.3	Analógové dataloggery	12
2.1.4	Digitálne dataloggery	12
2.2	Parametre a vlastnosti	13
2.2.1	Napájanie	13
2.2.2	Komunikačné rozhrania	13
2.2.3	Možnosti ukladania nameraných dát	14
2.2.4	Analógové a digitálne vstupy	14
2.2.5	Vlastnosti	14
3	Číslicové meranie analógových veličín	15
3.1	Analógovo - analógový prevodník (A/A)	15
3.2	Antialiasing filter (AAF)	15
3.3	Vzorkovač (VZ)	16
3.4	Analógovo-digitálny prevodník (A/D)	16
3.5	Digitálne-digitálny prevod (D/D)	16
4	Navrhnuté zariadenie	17
4.1	Riadiaca jednotka	17
4.1.1	Procesor	18
4.1.2	Pamäť	18
4.1.3	Sieťové pripojenie	18
4.1.4	I ² C zbernica	18
4.1.5	SPI rozhranie	18
4.1.6	GPIO	19
4.1.7	Napájanie	19
4.2	Pamäťová SD karta	19
4.3	Digitálne vstupy	19
4.4	Analógové vstupy	21
4.5	Napájanie dataloggera	23
4.6	Galvanické oddelenie SPI	24
4.7	Hodiny reálneho času	25
4.8	Ovládacie a signalizačné prvky	26
4.9	Pripojenie navrhnutého zariadenia k Raspberry Pi 2	26
5	Návrh dosky plošných spojov	27
6	Programové vybavenie	30
6.1	Inštalácia Raspberry Pi 2	30

6.1.1	Operačný systém	30
6.1.2	Inštalácia nástrojov	32
6.2	Hodiny reálneho času	33
6.3	Analógovo-digitálny prevodník MCP3204	37
6.4	Čítanie digitálnych vstupov	40
6.5	Databáza	41
6.6	Hlavná slučka	44
6.7	Webová stránka	47
6.8	Ovládanie dataloggeru	52
7	Test Funkčnosti.....	56
8	Záver.....	59
	Literatúra.....	61
	Zoznam skratiek a symbolov	63
	Zoznam príloh.....	64

Zoznam obrázkov

Obr. 2-1 Bloková schéma dataloggeru	13
Obr. 3-1 Proces prevedenia analógových veličín do digitálnej formy	15
Obr. 4-1 Bloková schéma navrhnutého dataloggera.....	17
Obr. 4-2 Raspberry Pi 2	18
Obr. 4-3 Oddelenie digitálnych vstupov.....	20
Obr. 4-4 Analógové vstupy.....	21
Obr. 4-5 Napäťový sledovač.....	21
Obr. 4-6 Antialiasing filter	22
Obr. 4-7 Zapojenie prevodníka.....	23
Obr. 4-8 Referenčný obvod	23
Obr. 4-9 DC/DC menič.....	24
Obr. 4-10 Lineárny stabilizátor 3,3 V.....	24
Obr. 4-11 Galvanické oddelenie SPI	25
Obr. 4-12 Hodiny reálneho času	25
Obr. 5-1 Doska plošných spojov - vrchná strana.....	27
Obr. 5-2 Doska plošných spojov – spodná strana.....	28
Obr. 5-3 Osadzovací plán súčiastok	29
Obr. 6-1 Win32DiskImager	30
Obr. 6-2 Konfiguračné nastavenia.....	31
Obr. 6-3 Program Putty.....	32
Obr. 6-4 Blokový diagram pc8563.....	34
Obr. 6-5 Blokový diagram AD prevodníka	37
Obr. 6-6 Princíp komunikácie prevodníka.....	38
Obr. 6-7 MySQL klient.....	42
Obr. 6-8 Vytvorená tabuľka.....	43
Obr. 6-9 Vývojový diagram.....	44
Obr. 6-10 Webová stránka nedefinovaná súborom <i>mystyle.css</i>	47
Obr. 6-11 Webová stránka definovaná súborom <i>mystyle.css</i>	48
Obr. 6-12 Rolovacie menu.....	49
Obr. 6-13 Vytvorená tabuľka v PHP	50
Obr. 6-14 Grafické zobrazenie nameraných hodnôt.....	52
Obr. 6-15 Nastavenia dataloggeru	53
Obr. 7-1 Tabuľka pre vstup DI_3	56
Obr. 7-2 Graf pre vstup DI_3	57
Obr. 7-3 Tabuľka pre vstup 0 - 10 V	57
Obr. 7-4 Graf pre vstup 0 - 10 V	58

Zoznam tabuliek

Tab. 6-1 Registre 0x02 - 0x08	34
Tab. 6-2 Riadiace bity.....	39

1 ÚVOD

Cieľom diplomovej práce je navrhnúť datalogger s ethernetovým rozhraním, ktorý bude zaznamenávať namerané dáta na pamäťovú kartu. V práci bude popísané všeobecné rozdelenie dataloggerov, ich vlastnosti a periférie. Budú objasnené štandardné vstupné signály a ich spracovanie pre použitie v dataloggeroch. Práca sa bude ďalej zaoberať návrhom elektrického zapojenia dataloggeru, výberom použitých komponentov a následne návrhom dosky plošných spojov. Navrhnutý datalogger bude mať naprogramované funkcie pre meranie vstupov. Namerané hodnoty budú zapísané do databázy.

V súčasnej dobe sa dataloggery používajú v priemysle ako diagnostické zariadenia pre sledovanie rôznych alarmov a porúch. Bývajú vybavené web servermi kvôli jednoduchému prístupu k nameraným údajom a ovládaniu v akomkoľvek čase z akéhokoľvek miesta.

2 DATALOGGER

Datalogger je elektronické zariadenie, ktoré slúži na zaznamenávanie údajov v čase, ktoré dostávame zo snímačov. Väčšinou je vybavený mikroprocesorom, údaje ukladá na pamäťové médium, ako napríklad SD karta. Ďalej obsahuje USB alebo Ethernet rozhranie, cez ktoré je pripojené do PC, displej na zobrazenie nameraných údajov a obslužné tlačidlá. Dataloggery bývajú kvôli prenosnosti napájané z batérií. Zaznamenávajú namerané hodnoty, či už analógových alebo digitálnych signálov.[1]

2.1 Rozdelenie dataloggerov

Dataloggery rozdeľujeme podľa rôznych kritérií na malé, veľké, jednoúčelové, viacúčelové, analógové a digitálne.

2.1.1 Jednoúčelový datalogger

Jednoúčelový datalogger slúži na zaznamenávanie dát z jednej meranej veličiny, ako sú napríklad teplota, vlhkosť, atď. Má pripojený iba jeden senzor a namerané hodnoty zobrazuje na displeji. Je vybavený niekoľkými tlačidlami na ovládanie. Namerané dáta ukladá do pamäti. Sú cenovo výhodnejšie ako univerzálne dataloggery. Patria ku nim kardiografy, termografy a barografy.[2]

2.1.2 Viacúčelový datalogger

Viacúčelový (univerzálny) datalogger zaznamenáva namerané hodnoty z viacerých senzorov. Je možné ku nemu pripojiť iba senzory, ktoré majú normalizovaný digitálny alebo analógový výstup, či už napäťový alebo prúdový.[2] Tento typ dataloggerov sa využíva najmä na diagnostiku v priemyselných systémoch.

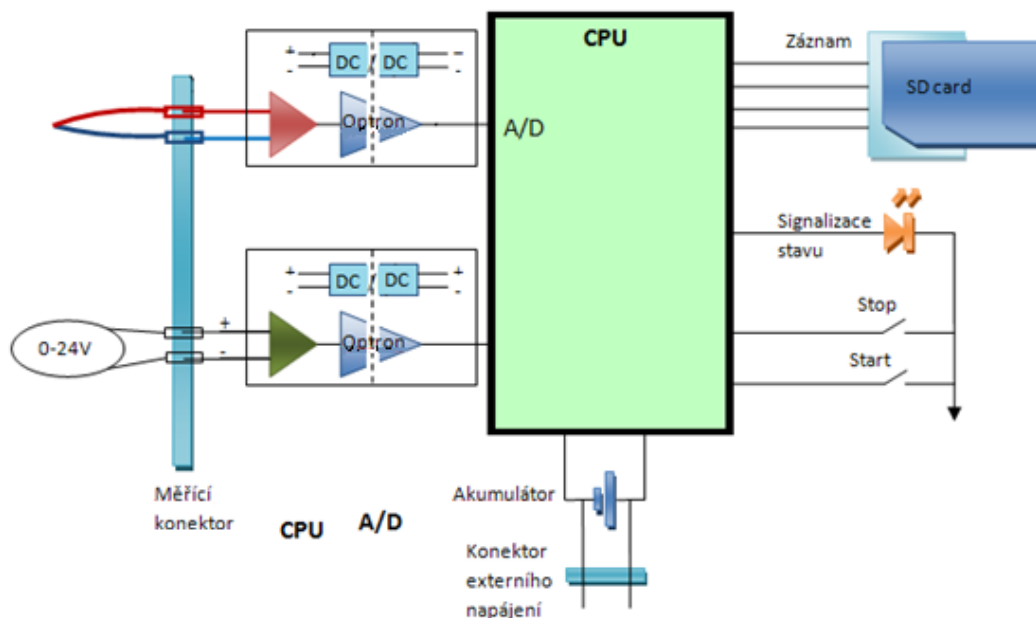
2.1.3 Analógové dataloggery

Analógové záznamníky (tzv. zapisovače) sú určené ku grafickému záznamu meranej veličiny. Pomenovanie zapisovače dostali vďaka ich funkcií zapisovať pomocou pera meranú veličinu na papier, pričom výchylka závisí od amplitúdy meranej veličiny v čase. Pracujú na mechanickom alebo elektromechanickom princípe[2]. V súčasnosti sú nahradené digitálnymi záznamníkmi.

2.1.4 Digitálne dataloggery

Číslicové elektronické záznamníky využívajú na záznam nameraných hodnôt pamäťové médium [3]. Sú vybavené rôznymi komunikačnými rozhraniami ako napríklad RS232, USB, ETHERNET. Pre riadenie celého systému sa používajú mikroprocesory. Sú modernejším typom analógových dataloggerov a v súčasnosti sú využívanéjšie. Na rozdiel od analógových sú vybavené displejom, ktorý nahrádza pero

a papier. Ich výhodou je, že sú malé, prenosné a pre napájanie používajú akumulátory. Majú viac vstupov, preto je možné zaznamenávať viacero meraných veličín. Obsahujú hodiny reálneho času, z ktorých sa robí záznam času, kedy bola veličina meraná. Späťne je možné z týchto vzoriek získať grafický priebeh meranej veličiny v závislosti na čase [2].



Obr. 2-1 Bloková schéma dataloggeru [4]

2.2 Parametre a vlastnosti

2.2.1 Napájanie

Dataloggery majú pre napájanie pripojené batérie kvôli prenosnosti a pre ich nízku spotrebu. Sú to nabíjacie akumulátory typu AA Li-Ion, Lipol, ktoré sa využívajú kvôli ich veľkej kapacite a malému samovybíjaniu. Zariadenie obsahuje aj nabíjačku na tieto batérie. Veľkosť napájacieho napätia sa pohybuje v rozsahu 5-24V. Je možné ku nim pripájať externé jednosmerné napájacie zdroje.

2.2.2 Komunikačné rozhrania

Pre prepojenie zariadení s počítačom sa používajú rôzne komunikačné rozhrania. Pri jednoduchých dataloggeroch hovoríme o základných komunikačných rozhraniach, ako USB, a u starších, sériové linky RS232 a RS485. Zložitejšie dataloggery sú už vybavené priemyselnými rozhraniami ako CAN, Ethernet. Najrozšírenejšie je USB komunikačné rozhranie.

2.2.3 Možnosti ukladania nameraných dát

Dáta merané po nejakú dobu je potrebné zálohovať. V súčasnosti sú veľmi obľúbené SD pamäťové karty s pamäťou flash. Pri dostupných veľkostiach pamäte (4GB) je možné uchovávať až niekoľko miliónov vzoriek. Dĺžka záznamu závisí od periódy vzorkovania, čo v niektorých prípadoch môže byť dlhý záznam až jeden rok.

2.2.4 Analógové a digitálne vstupy

Od zložitosti dataloggerov závisí počet analógových vstupov. Niektoré dataloggery môžu byť vybavené až 32 analógovými vstupmi. Analógové vstupy sú najčastejšie používané v rozsahu 0-10V, 0-5V a 4-20mA. Tieto vstupy musia byť upravené tak, aby vyhovovali vstupnému signálu pre A/D prevodník. Pre prúdovú slučku 4-20mA sa väčšinou používa prevodník prúdu na napätie alebo meranie prúdu pomocou napätia na rezistore. Takto upravené napätie privádzame na A/D prevodníky so štandardným rozsahom 12-16 bitov, kde dostaneme veľkosť napätia v digitálnom tvare. Všetky tieto analógové vstupy musia byť galvanicky oddelené. [2] Najčastejšie sa pripájajú senzory teploty, tlaku a vlhkosti.

Digitálne vstupy sa bežne používajú v priemysle v počtoch 4, 8, 16, 32, pričom ich napätie je 24V. Taktiež aj tieto vstupy musia byť galvanicky oddelené, často používaným oddelením je optočlen, kombinácia diódy a fototranzistora.

2.2.5 Vlastnosti

Modernejšie dataloggery bývajú vybavené rôznymi doplnkami a s väčším počtom vstupov, či už analógových alebo digitálnych.

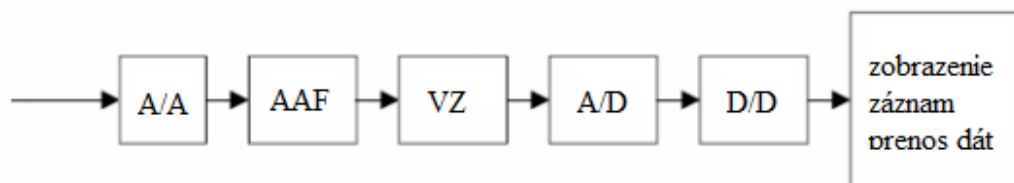
Jedným z nich je bezdrôtové pripojenie na sieť. Je tiež možná online komunikácia z rôznych miest vďaka webovému serveru v zariadení, pripojeniu na internet a počítaču. Používajú sa často v diagnostike na hlásenie rôznych alarmov a porúch.

Ďalším doplnkom je GPRS modem, ktorým je možné pripojiť sa na mobilné siete.

Obsahujú tiež grafické displeje, ktoré slúžia na vykreslenie hodnôt v čase priamo na zariadení, prípadne na nastavenie parametrov zariadenia.

3 ČÍSLICOVÉ MERANIE ANALÓGOVÝCH VELIČÍN

Ak chceme analógové veličiny ďalej spracovávať je vhodné ich previesť do digitálnej formy, čo realizujeme prostredníctvom šiestich nasledujúcich krokov (obr. 3-1).



Obr. 3-1 Proces prevedenia analógových veličín do digitálnej formy

3.1 Analógovo - analógový prevodník (A/A)

Analógová veličina, ktorú chceme merať, nie je vždy vyhovujúca pre A/D prevodník, práve preto je potrebné ju upraviť na inú analógovú veličinu s menším alebo s väčším napäťovým rozsahom. A to tak, aby to vyhovovalo vstupnému rozsahu A/D prevodníku [3]. Niekedy je však potrebné previesť jednu elektrickú veličinu na druhú, napríklad prúd na napätie alebo napätie na frekvenciu. Pri A/A prevodníku musia byť dodržané dynamické vlastnosti v závislosti na meranom signáli. Prevodník môže často ovplyvniť offset meraného signálu prípadne jeho nelinearitu. V praxi sa často používajú prevodníky napäťových úrovní alebo prevodníky jednej elektrickej veličiny na inú elektrickú veličinu. Pri prevádzaní vstupného signálu, ktorým je elektrické napätie, nám zvyčajne stačí meraný vstupný signál zosilniť alebo zoslabiť.

3.2 Antialiasing filter (AAF)

Antialiasing filter sa používa pred vzorkovačom, aby obmedzil prienik vyšších harmonických vzoriek, ktoré nevyhovujú vzorkovaciemu teorému pre A/D prevodník, ktorý hovorí že vzorkovacia frekvencia musí byť dvakrát väčšia ako najvyššia frekvencia meraného signálu. Pri nedodržaní vzorkovacieho teorému môže nastať chyba merania. [3] V praxi sa často používa vyššia vzorkovacia frekvencia ako je dvojnásobok meranej frekvencie. Antialiasing filter má za úlohu tieto vyššie harmonické zložky potlačiť tak, aby sa na vstupe A/D prevodníku prejavovali čo najmenej a aby zároveň boli menšie ako kvantizačná chyba A/D prevodníka. Nevýhodou antialiasing filtra je

deformácia fázovej charakteristiky spektra, následkom čoho dochádza k zníženiu presnosti merania. [3]

3.3 Vzorkovač (VZ)

Vzorkovač, ktorý chápeme ako analógovú pamäť, je umiestnený hneď za antialiasing filtrom. Tento blok má za úlohu uchovávať veľkosť meranej analógovej veličiny v okamihoch vzorkovania. Vzorkovaním je rozdelená meraná veličina v čase (kvantovanie) [3]. Toto vzorkovanie by malo prebehnúť v čase blížiacom sa nule, čo nie je možné zrealizovať a tak vzniká chyba, pričom nevieme, aká hodnota veličiny je uložená vo vzorkovači, či počiatočná, koncová alebo priemerná. Keďže vzorkovač býva často súčasťou A/D prevodníkov, jeho vlastnosti sú zahrnuté v parametroch A/D prevodníka.

3.4 Analógovo-digitálny prevodník (A/D)

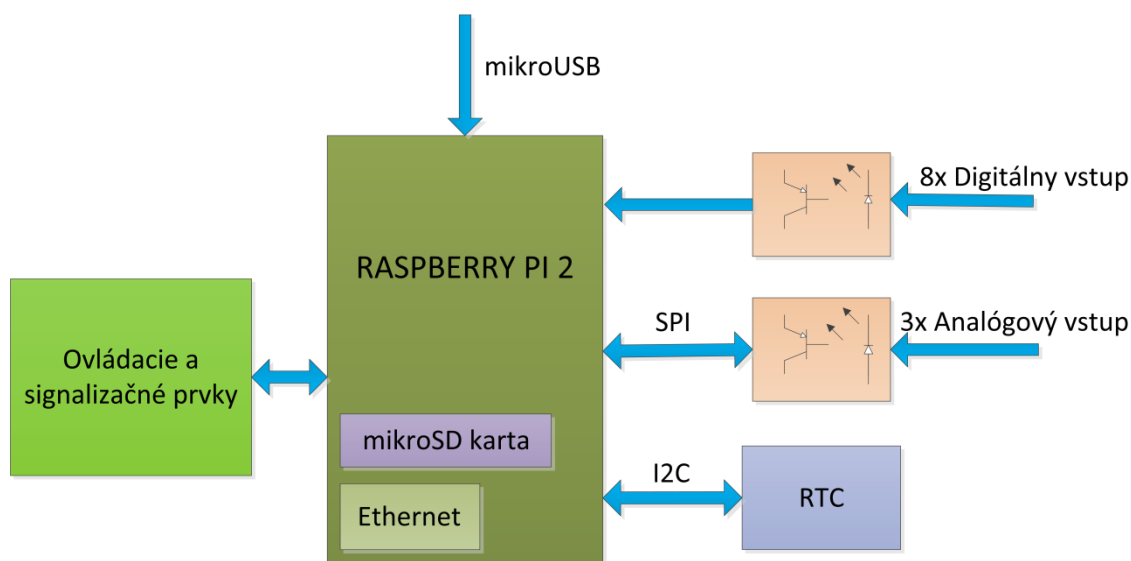
Samotný analógovo-digitálny prevodník nám prevádza navzorkované napätie zo vzorkovača na digitálny signál vyjadrený v binárnej kombinácii. Medzi najpoužívanéjšie typy A/D prevodníkov patria sigma-delta, prevodník s postupnou aproximáciou a prevodníky s dvojitou integráciou, kde každý typ používame na prevod iných elektrických veličín. Dôležitými parametrami A/D prevodníka sú rozlíšenie, maximálna frekvencia vzorkovania, napájacie napätie a druh digitálneho rozhrania.

3.5 Digitálne-digitálny prevod (D/D)

Za A/D prevodníkom nasleduje blok digitálne- digitálny prevod, ktorým spracovávame výsledok z A/D prevodníka a prevádzame to na inú digitálnu hodnotu pomocou matematických operácií [3]. Najčastejšie upravujeme zosilnenie, offset a filtráciu. Výsledné hodnoty meraného signálu nemusia byť presné, pretože výpočty sa realizujú s konečnou presnosťou, tzn. pevná alebo plávajúca rádova čiarka.

4 NAVRHNUTÉ ZARIADENIE

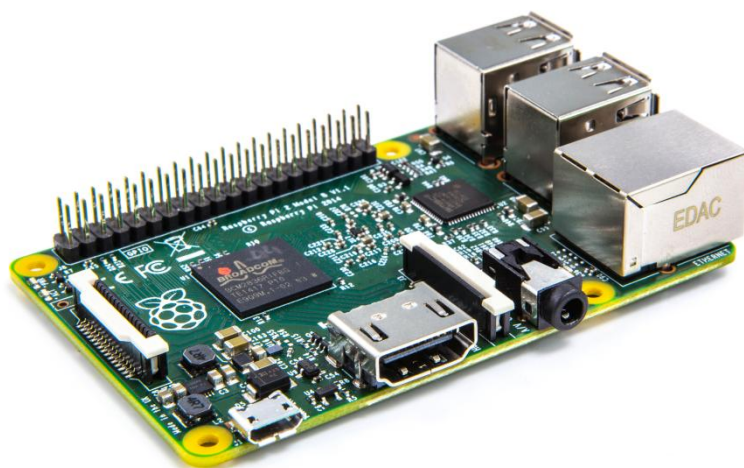
Datalogger, ktorý sme navrhli sa skladá z niekoľkých základných blokov. Nevyhnutnou súčasťou každého dataloggeru je riadiaca jednotka, úložisko nameraných hodnôt, ktoré predstavuje pamäťová SD karta, komunikačné rozhranie, napájací zdroj a samotné analógové a digitálne vstupy. Blokovú schému je možné vidieť na obr. 4-1.



Obr. 4-1 Bloková schéma navrhnutého dataloggera

4.1 Riadiaca jednotka

Pre náš datalogger bude ako riadiaca jednotka slúžiť Raspberry Pi 2 model B od výrobcu Raspberry Pi Foundation. Patrí medzi mikropočítače s veľkosťou kreditnej karty. Obsahuje všetko, čo by mal počítač mať pre svoju činnosť. Má v sebe zabudovanú mikroSD kartu, komunikáciu pomocou ethernetu, rozhrania SPI a I2C, čo je pre náš datalogger potrebné. V tomto čase je už najnovšia verzia Raspberry Pi 3 model B. Podľa technických parametrov by nám vyhovovala aj staršia verzia, ale ako školská pomôcka nám bola poskytnutá verzia 2.[5]



Obr. 4-2 Raspberry Pi 2 [7]

4.1.1 Procesor

Raspberry Pi 2 je vybavený čipom BCM2835 od firmy Broadcom, ktorý obsahuje 32-bitový štvorjadrový procesor ARM Cortex-A7 s frekvenciou 900 MHz.[5]

4.1.2 Pamäť

Verzie Raspberry Pi 2 a 3 majú až 1 GB operačnej pamäte RAM. Prvé vydanie má iba 512 MB.[5]

4.1.3 Sieťové pripojenie

Raspberry Pi má dve možnosti pripojení do siete. Prvá pomocou ethernetového rozhrania cez konektor RJ – 45 a druhá pomocou Wifi Dongle. Náš datalogger bude využívať káblové spojenie, tak bude pripojený do siete cez ethernetový konektor.

4.1.4 I²C zbernica

Táto zbernica patrí medzi štandardnú výbavu mikroprocesorov. Používa sa na pripojenie rôznych periférií, ako napríklad displeje, pamäte a prevodníky. Vyvinula ju firma Philips Semiconductors. Jedná sa o dvojvodičovú zbernicu typu Master – Slave s vodičmi SDA (pre dáta) a SCL (pre hodiny). V Raspberry Pi 2 sa tieto signály nachádzajú na pinoch 2 a 3. Datalogger bude túto zbernicu využívať pre komunikáciu s externými hodinami reálneho času.[5]

4.1.5 SPI rozhranie

SPI (Serial Peripheral Interface) patrí medzi synchrónne sériové komunikačné rozhrania. Bolo vyvinuté firmou Motorola a tiež je súčasťou štandardnej výbavy pre mikroprocesory. Komunikácia je založená na type Master – Slave. Fyzická vrstva obsahuje štyri signály MOSI (dáta Master → Slave), MISO (dáta Slave → Master),

SCLK (synchronizačné hodiny) a SS (riadiaci signál pre komunikáciu s daným Slave). Raspberry Pi 2 má zabudované SPI na pinoch 19, 21, 23 pre signály MOSI, MISO a SCLK. Na výber sú až dva piny signálu SS (piny 24 a 26). My budeme používať SS na pine 24. Toto rozhranie je často využívané na komunikáciu napríklad s pamäťovými kartami, AD prevodníkmi, atď.[5]

4.1.6 GPIO

Na pripojenie signálov, či už vstupov alebo výstupov má Raspberry Pi 2 pinovú lištu 2x20 pinov. Tieto piny sú nazvané GPIO (General Purpose Input Output) piny. Dajú sa použiť ako vstupy a výstupy s napäťovými úrovňami 3,3 V pre logickú jednotku a 0 V pre logickú nulu, alebo ich môžeme využiť pre alternatívne funkcie, ako vyššie spomínané SPI alebo I²C. V dataloggeri budú niektoré piny využité na pripojenie digitálnych vstupov. Týmto pinom sa dajú interne nastaviť pull-up/down rezistory.[5]

4.1.7 Napájanie

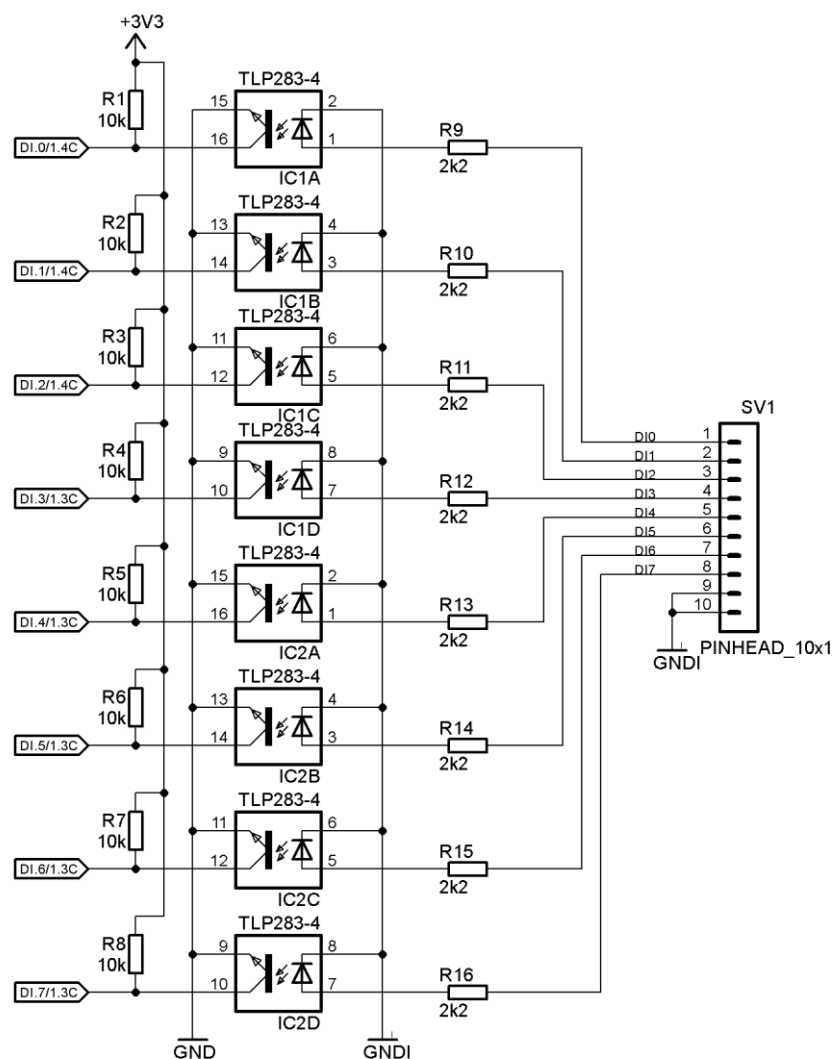
Zariadenie Raspberry Pi je napájané 5V mikro USB adaptérom. Typická spotreba je v rozmedzí 700 – 1000 mA. Spotreba je závislá na tom, čo všetko máme na Raspberry Pi pripojené, a čo všetko používame.

4.2 Pamäťová SD karta

Ako úložisko nameraných dát je použitá pamäťová karta typu Multimedia Card a Secure Digital rozmer mikro SD. Je to spoločná karta aj pre operačný systém v Raspberry Pi. Odporúčaná veľkosť karty je 8 GB. Operačný systém zaberá približne 1,5 GB a zostatok môžeme použiť na namerané dáta, ktoré budeme zapisovať do databázy.[5]

4.3 Digitálne vstupy

Datalogger je navrhnutý na vstupné digitálne signály s veľkosťou napätia 24V. Počet týchto signálov je 8. Signály sú galvanicky oddelené od riadiacej jednotky pomocou optočlenov (kombinácia LED dióda a fototranzistor) ako môžeme vidieť na obrázku 4-3.



Obr. 4-3 Oddelenie digitálnych vstupov

Oddelovacie obvody IC1 a IC2 TLP 283-4 sú od firmy TOSHIBA, tvoria ich štyri optočleny. Z dokumentácie týchto obvodov boli použité základné údaje ako optimálny prúd prechádzajúci LED diódou a úbytok napätia na LED dióde. Pomocou týchto údajov bolo potrebné vypočítať odpor sériovo pridaný k LED dióde. Izolačné napätie obvodov je 2,5kV.[9]

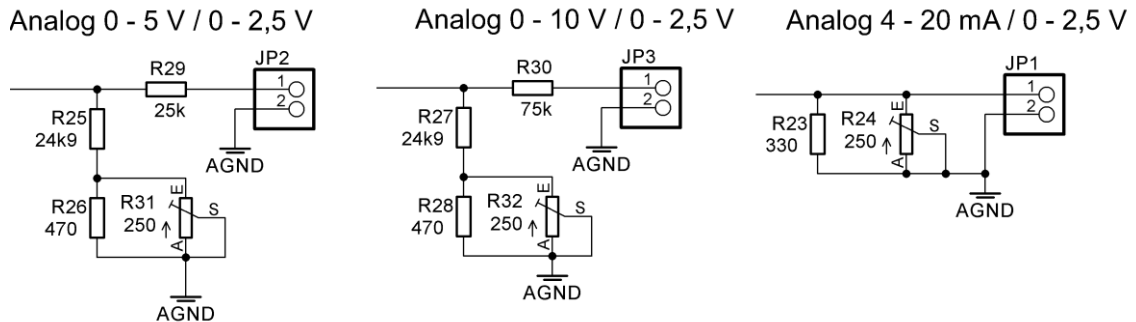
$$R_{9-16} = \frac{U_{DI} - U_D}{I_f} = \frac{24 - 1,15}{0,01} = 2\,285\,\Omega \quad (4.1)$$

kde U_{DI} odpovedá vstupnému napätiu z digitálneho vstupu, U_D úbytok napätia na LED dióde a I_f prúd LED diódou.

K riadiacej jednotke sú oddelené vstupy pripojené signálmi DI.0 až DI.7. K týmto signálom sú pripojené pull – up rezistory k zaisteniu logickej úrovne.

4.4 Analógové vstupy

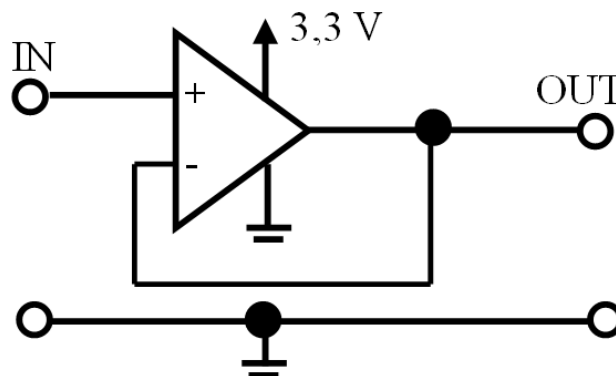
V zariadení boli navrhnuté tri analógové vstupy v štandardných rozsahoch. Dva vstupy sú napäťové 0-5V, 0-10V a jeden prúdový 4-20mA. Napäťové vstupné signály sú upravené odporovými deličmi na veľkosť napätia potrebnú pre rozsah AD prevodníka. Prúdový vstup je prevedený na napätie pomocou bočníka.



Obr. 4-4 Analógové vstupy

Pre prúdový vstup bol vypočítaný bočník a to tak, aby hodnota 20 mA odpovedala napätiu 2,5V. U napäťových vstupných signáloch bol navrhnutý odporový delič, pre signál 0 - 10 V v pomere 3:1 (R30, R27) a pre signál 0 - 5 V v pomere 1:1 (R29, R25). Kvôli presnému nastaveniu deličov sú do vetiev pridané trimre R32 a R31 paralelne spojené s rezistormi R28 a R26.

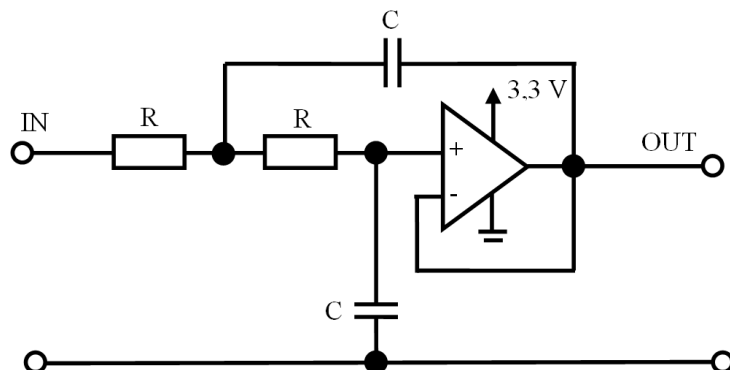
Takto upravené analógové vstupy, ktorých rozsah by nemal presahovať hodnotu napätia 2,5 V, sú privedené na napäťový sledovač (obr. 4-5).



Obr. 4-5 Napäťový sledovač

Ide o jednoduchý elektronický obvod, ktorý tvorí operačný zosilňovač. Invertujúci vstup má priamo pripojený na výstup, čo zabezpečuje, že výstup nie je zosilnený ani zoslabený. Použitý je kvôli vysokej vstupnej impedancii, čím čo najmenej zaťažuje použitý analógový vstup.

Za týmto obvodom nasleduje obvod antialiasing filter. Ten sa používa kvôli dodržaniu vzorkovacieho teorému, ako je už spomínané vyššie. Vybraný bol jeden z jednoduchých dolno–priepravných filtrov Butterworth (obr. 4-6).



Obr. 4-6 Antialiasing filter

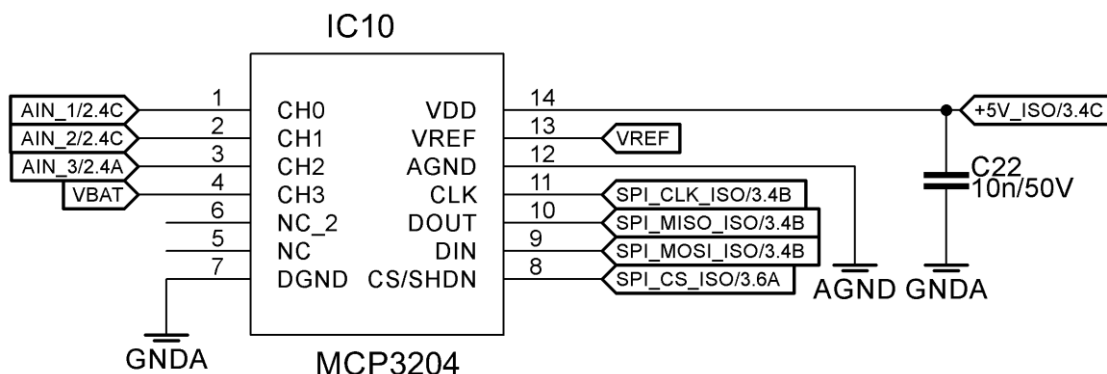
Výpočet prvkov R a C pochádza zo vzťahu

$$f_c = \frac{1}{2\pi RC}, [Hz] \quad (4.2)$$

kde f_c odpovedá 70% polovice vzorkovacej frekvencie, v našom prípade 35 kHz, veľkosť R sme zvolili 100 k Ω a vypočítaná veľkosť kapacity kondenzátorov vyšla 47 nF.

Realizácia sledovačov a filtrov bola pomocou integrovaných obvodov od firmy Texas Instruments TLV2764 a TLV2762 [10]. Obvody boli vybrané na základe nesymetrického napájania 3,3 V a veľmi nízkej spotrebe 20 μ A na kanál.

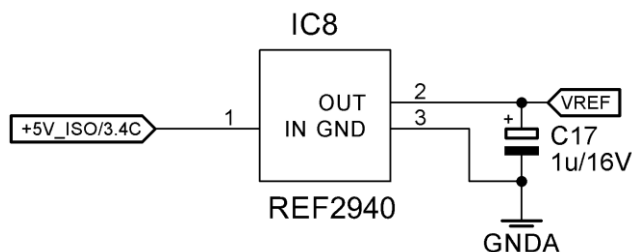
Výstupy z filtrov sú privedené na vstupy AD prevodníkov. Pre datalogger bol vybraný AD prevodník MCP3204 od firmy Microchip. Prevodník má nízku spotrebu približne 15,5 mW. Rozlíšenie prevodníka je 12 bitov, čo je pre datalogger postačujúce. Vzorkovacia frekvencia týchto prevodníkov je pri napájaní 2,7 V 50 kHz a pri 5 V 100 kHz. Napájanie sa môže pohybovať v rozmedzí 2,7 V – 5,5V [11]. Metóda prevodu analógového signálu na digitálny tohto prevodníka je postupná aproximácia. AD prevodník môžeme použiť na prevádzanie až štyroch signálov, pretože prevodník obsahuje na vstupe mutliplexor, kde si programovo volíme, ktorý vstup budeme merať. Takže jedným prevodníkom prevádzame napätia 0-5 V, 0-10V, prúdovú slučku 4-20 mA a napätie na RTC batérii. Komunikácia AD prevodníka je formou sériového rozhrania SPI.



Obr. 4-7 Zapojenie prevodníka

Prevodník je napájaný 5 V. Na vstupy prevodníka IC10 sú privedené napäťové signály AIN_1 – AIN_3, ktoré odpovedajú analógovým vstupom a napätie RTC batérie VBAT. Zapojenie prevodníka pochádza z dokumentácie k tomuto prevodníku [11]. Komunikačné a riadiace signály SPI rozhrania sú cez galvanické oddelenie privedené k SPI riadiacej jednotky.

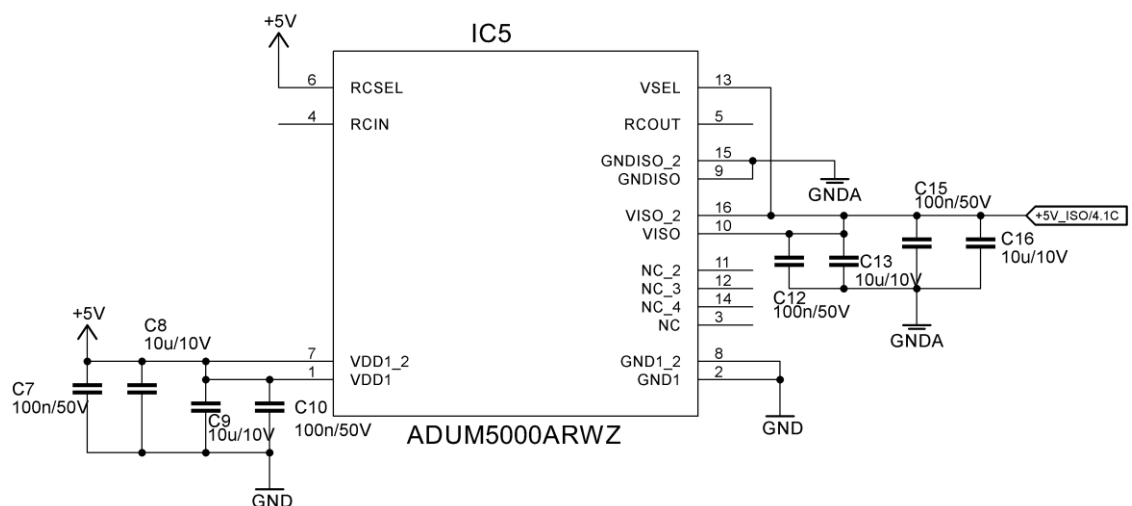
Na referenčné napätie bol použitý nízko - energetický obvod IC8 REF2940 od Texas Instruments, ktorého katalógové zapojenie [12] môžeme vidieť na obr. 4-8.



Obr. 4-8 Referenčný obvod

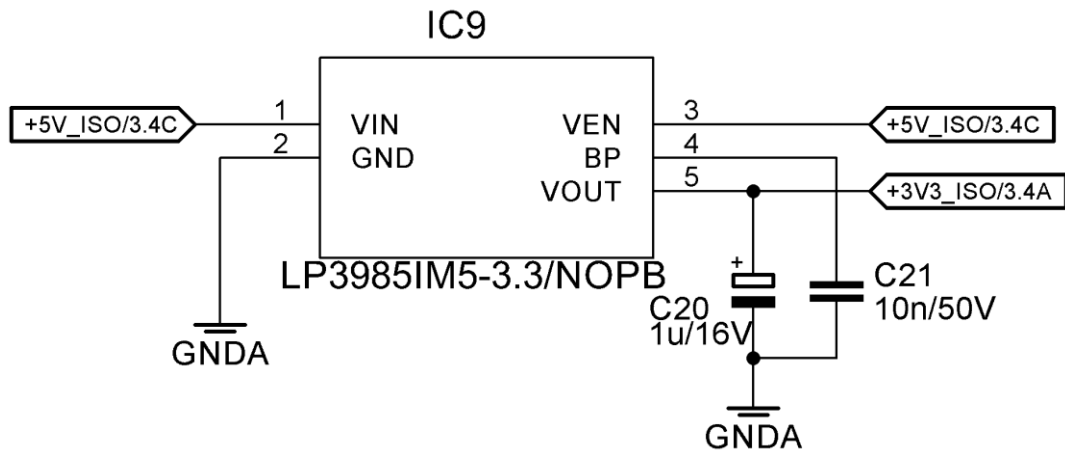
4.5 Napájanie dataloggera

Datalogger je napájaný z napätia, ktoré poskytuje Raspberry Pi. Na výber sú dve napätia, a to 3,3 V a 5 V. V digitálnej neoddelennej časti využívame obe tieto napätia, napríklad ako napájanie pre LED diódy alebo hodín reálneho času. Napätie 5 V z Raspberry Pi je kvôli ďalšiemu použitiu pre obvody týkajúce sa analógových vstupov a ich úpravy galvanicky oddelené DC/DC meničom. Použitý je ADUM5000ARWZ od firmy Analog Devices [17]. Izolačné napätie tohto meniča je 2,5 kV. Jeho zapojenie môžeme vidieť na obr. 4-9.



Obr. 4-9 DC/DC menič

Takto galvanicky oddelené napätie 5V_ISO ďalej potrebujeme zmenšiť na 3,3 V. Na to je vhodný stabilizátor IC9 LP3985IM5 od firmy Texas Instruments. Zmenšeným napätím ďalej poskytujeme napájanie obvodom, ktoré používame v oddelenej analógovej časti. Na obr. 4-10 môžeme vidieť zapojenie, ktoré pochádza z dokumentácie k tomuto obvodu [16].

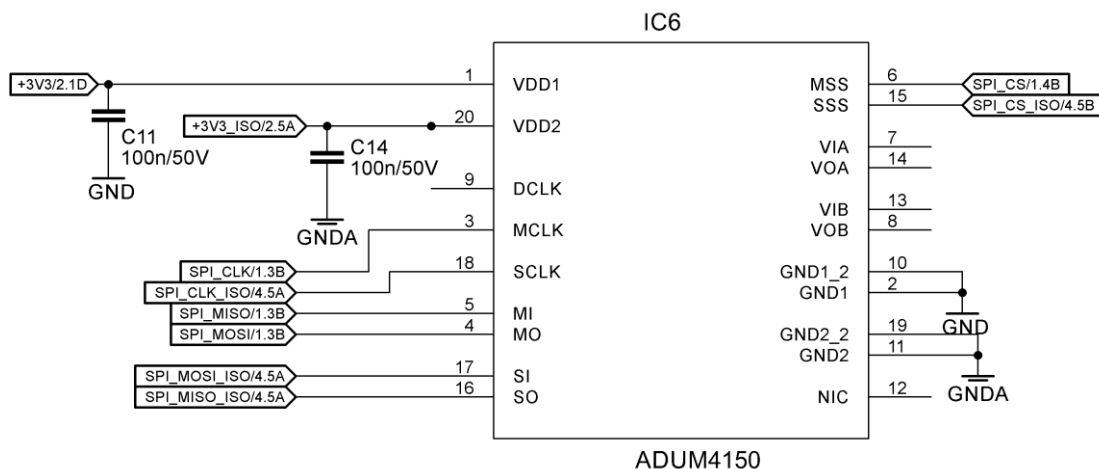


Obr. 4-10 Lineárny stabilizátor 3,3 V

4.6 Galvanické oddelenie SPI

Pre úplné oddelenie digitálnej a analógovej časti je ešte potrebné oddeliť aj komunikáciu AD prevodníka s Raspberry Pi. Na tieto účely sú určené špeciálne obvody pre každý druh komunikácie. Konkrétne pre nás je vhodný obvod IC6 ADUM4150 od

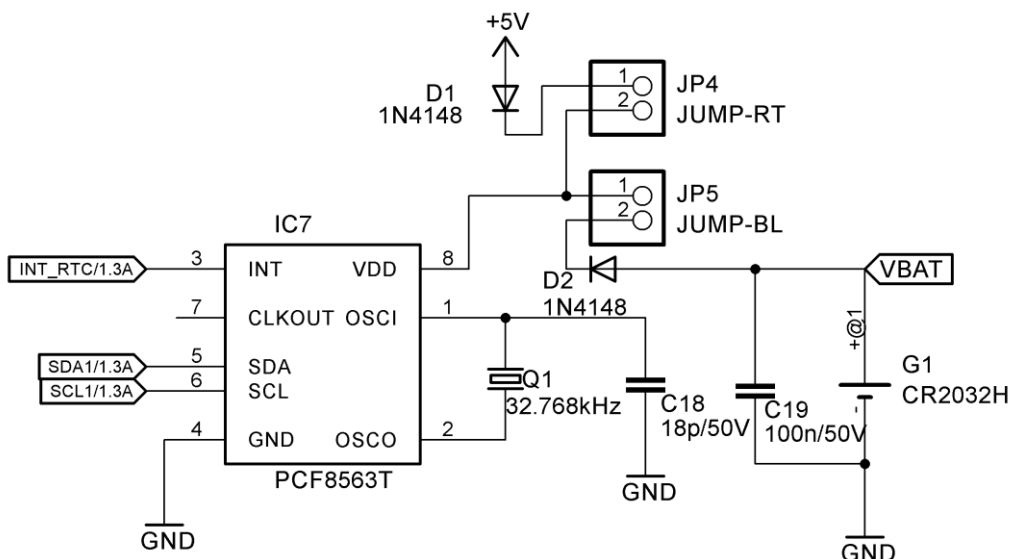
firmy Analog Devices s izolačným napätím 5 kV. Zapojenie podľa dokumentácie môžeme vidieť na obr.4-11. [6]



Obr. 4-11 Galvanické oddelenie SPI

4.7 Hodiny reálneho času

Keď z Raspberry Pi odpojíme napájanie, tak v ňom prestanú fungovať interné hodiny. Namerané hodnoty v dataloggeri by mali mať údaj o čase, kedy boli tieto hodnoty zmerané. Aby sme predišli tomuto problému s nesprávnym údajom o čase, vybavili sme datalogger obvodom hodín reálneho času. Ide o jednoduchý obvod PCF8563T [13] od firmy NXP Semiconductor obsahujúci hodiny a kalendár. Zapojenie obvodu je vidieť na obr. 4-12.



Obr. 4-12 Hodiny reálneho času

Obvod je pripojený k Raspberry Pi cez zbernicu I²C signálmi SDA a SCL. Ako oscilátor je použitý presný kryštál o frekvencii 32,786 kHz spolu s kondenzátorom C18. Veľkosť kapacity kondenzátora by mala byť v rozmedzí 10 – 20 pF. Po pripojení na osciloskop sa kapacita 18 pF prejavila ako najvhodnejšia, lebo pri nej bola frekvencia

prerušení 1 Hz najpresnejšia. Kvôli výpadku napájania je k obvodu pridaná batéria, aby hodiny neprestali ísť. Diódy D1 a D2 slúžia na to, aby batéria napájala iba tento obvod. Pre potrebu vymazania týchto hodín slúžia prepajky JP4 a JP5.

4.8 Ovládacie a signalizačné prvky

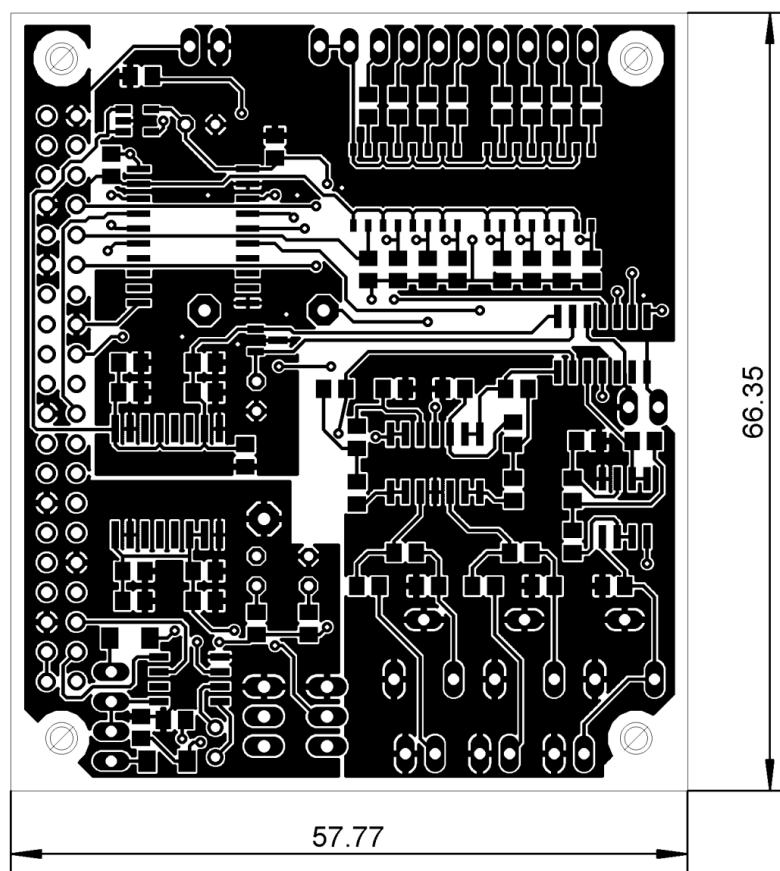
Datalogger má navrhnuté dva prepínače S1 a S2, ktoré slúžia na odpojenie napätí 5 V a 3 V privedené z Raspberry Pi. V zapojení sa nachádzajú aj dve prídavné LED diódy, ktoré slúžia na signalizáciu stavov dataloggeru. LED1 na signalizáciu stavu merania a LED2 stavu zapnutia dataloggeru.

4.9 Pripojenie navrhnutého zariadenia k Raspberry Pi 2

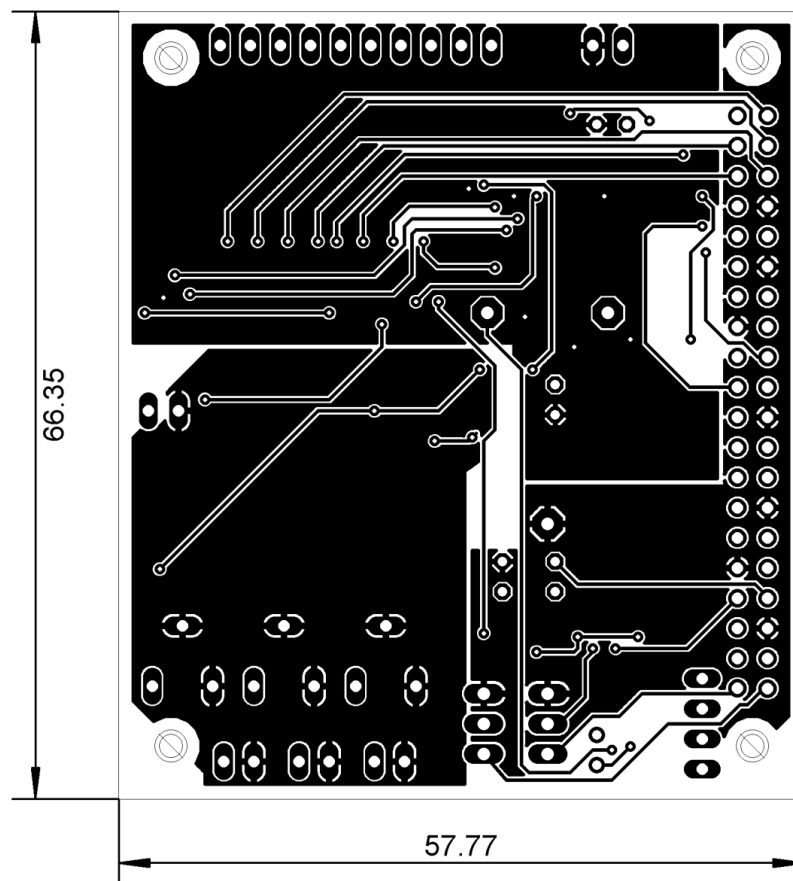
Navrhnuté zariadenie je pripojené k Raspberry Pi 2 pomocou štyridsať-vodičového plochého kábla. Na koncoch tohto kábla sú pripojené 2x20 pinové konektory s roztečou 2,54 mm. Použité nie sú všetky piny, len niektoré, cez ktoré sú privedené potrebné signály k riadiacej jednotke.

5 NÁVRH DOSKY PLOŠNÝCH SPOJOV

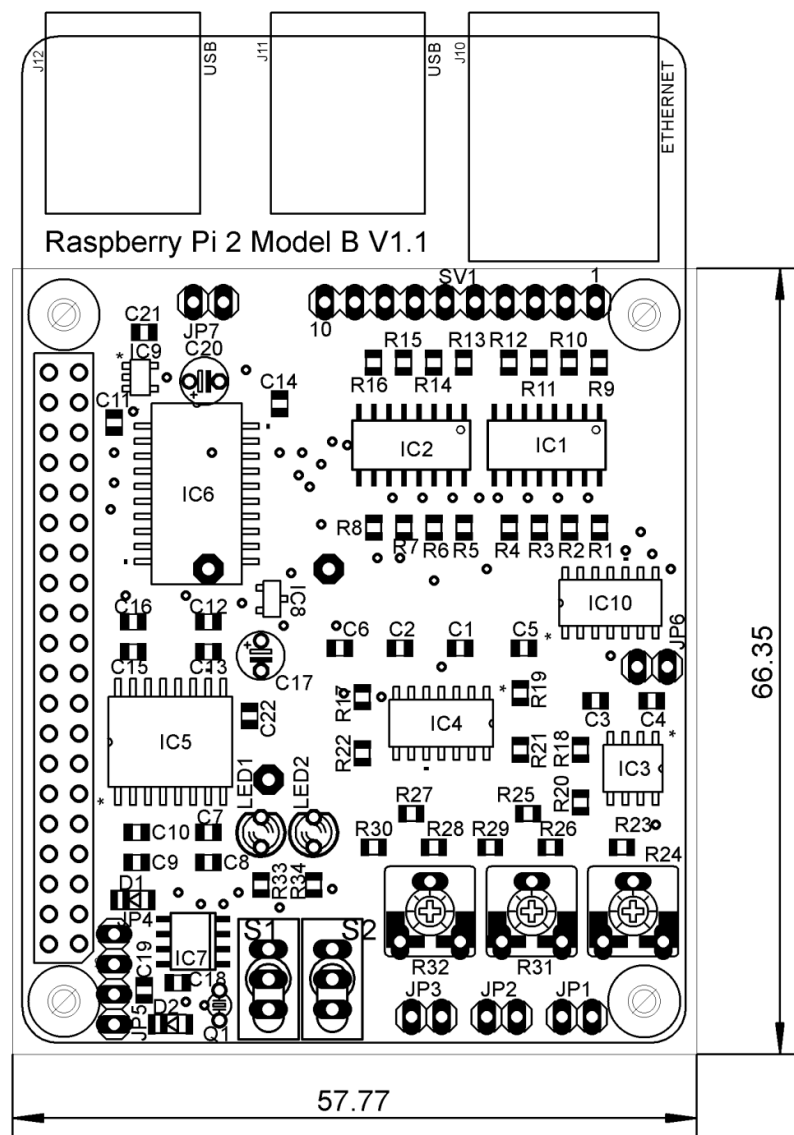
Plošný spoj pre datalogger bol navrhovaný v programe Eagle. Doska má rozmery 57,77 x 66,35 mm. Tento rozmer je zvolený podľa rozmeru Raspberry Pi 2 tak, aby doska dataloggeru nebola väčšia. Bola zvolená jednostranná SMD montáž. Na vrchnej strane sú umiestnené všetky súčiastky, prepínače, signalizačné LED a konektory vstupov. Na spodnej strane je umiestnená len batéria pre hodiny reálneho času kvôli nedostatku miesta na vrchnej strane.



Obr. 5-1 Doska plošných spojov - vrchná strana



Obr. 5-2 Doska plošných spojov – spodná strana



Obr. 5-3 Osadzovací plán súčiastok

6 PROGRAMOVÉ VYBAVENIE

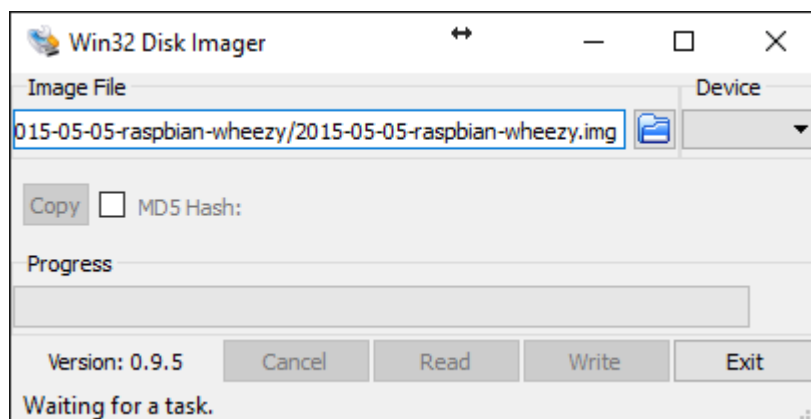
Programové vybavenie dataloggeru sa skladá z dvoch častí. Prvá časť sa zaoberá programami na obsluhu periférnych obvodov pripojených k riadiacej jednotke Raspberry Pi 2. Tieto programy sú napísané v jazykoch C a Python. Programy v druhej časti sú zamerané na prácu s databázami a webovým serverom pomocou značkovacieho jazyka HTML, skriptovacích jazykov PHP a JavaScript, a jazykom pre prácu s databázami MySQL. Tieto jazyky boli vybraté na základe predchádzajúcich skúseností.

6.1 Inštalácia Raspberry Pi 2

Pri prvom spustení Raspberry Pi, sme potrebovali mikro SD kartu, monitor pripojený cez HDMI konektor, klávesnicu a zdroj napätia. Druhá možnosť pripojenia je pomocou ethernetového kábla, kde na smerovači musí byť zapnuté DHCP. Na Raspberry Pi je DHCP klient štandardne zapnutý. Toto spojenie bude fungovať len v takomto prípade, inak by Raspberry Pi nebola priradená IP adresa.

6.1.1 Operačný systém

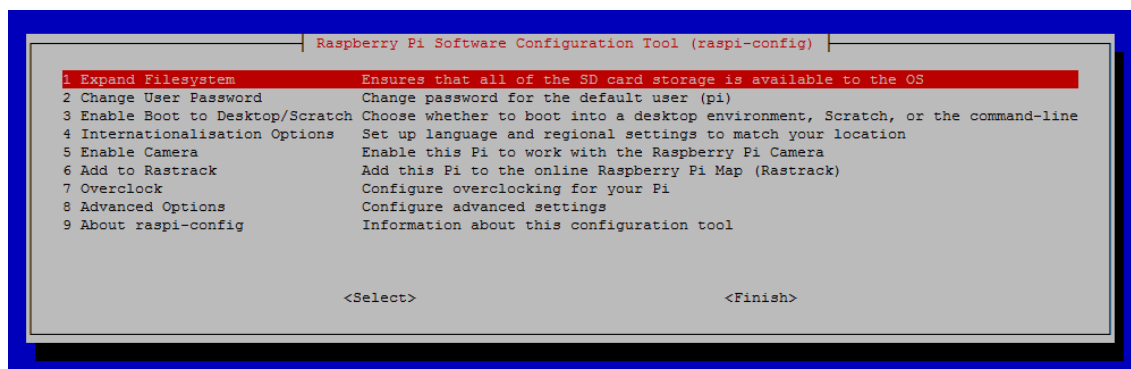
Raspberry Pi podporuje viacero operačných systémov, no najčastejšie používaný je operačný systém Raspbian. Jednou z možností ako tento operačný systém nainštalovať je nahráť ho na mikro SD kartu. Karta musí mať minimálnu kapacitu 4 GB a byť aspoň Class 4 triedy. Pre naše Raspberry bola poskytnutá mikro SD karta s kapacitou 8 GB triedy Class 10. Operačný systém Raspbian je voľne stiahnuteľný na webových stránkach k Raspberry Pi [14]. Stiahnutý skomprimovaný súbor je ďalej potrebné rozbaľiť a rozbaľený obraz vypáliť na kartu. Jeden z vhodných voľne dostupných programov podporovaných vo Windowse je aj Win32DiskImager.



Obr. 6-1 Win32DiskImager

Po výbere správneho zariadenia (s pamäťovou kartou) a obrazu operačného systému sme zapísali pomocou tlačidla *write* obraz na kartu. Takto nahratú kartu je možné vložiť do Raspberry Pi a zapnúť napájanie.

V úvode po nabehnutí systému je potrebné zadať prihlasovacie údaje. Štandardné meno je „pi“ a heslo „raspberry“. Tieto prihlasovacie údaje sa dajú zmeniť v konfiguračných nastaveniach Raspberry Pi. Konfiguračné nastavenia sme spustili zadáním príkazu `sudo raspi-config`. Na obrazovke sa nám zobrazili nastavenia, ktoré môžeme vidieť na obr. 6-2.



Obr. 6-2 Konfiguračné nastavenia

V ďalšom kroku sme povolili rozšírenie celej kapacity pamäťovej karty pre operačný systém, pretože štandardne je to obmedzené. Týmto rozšírením sme zabezpečili možnosť väčšieho počtu nameraných dát.

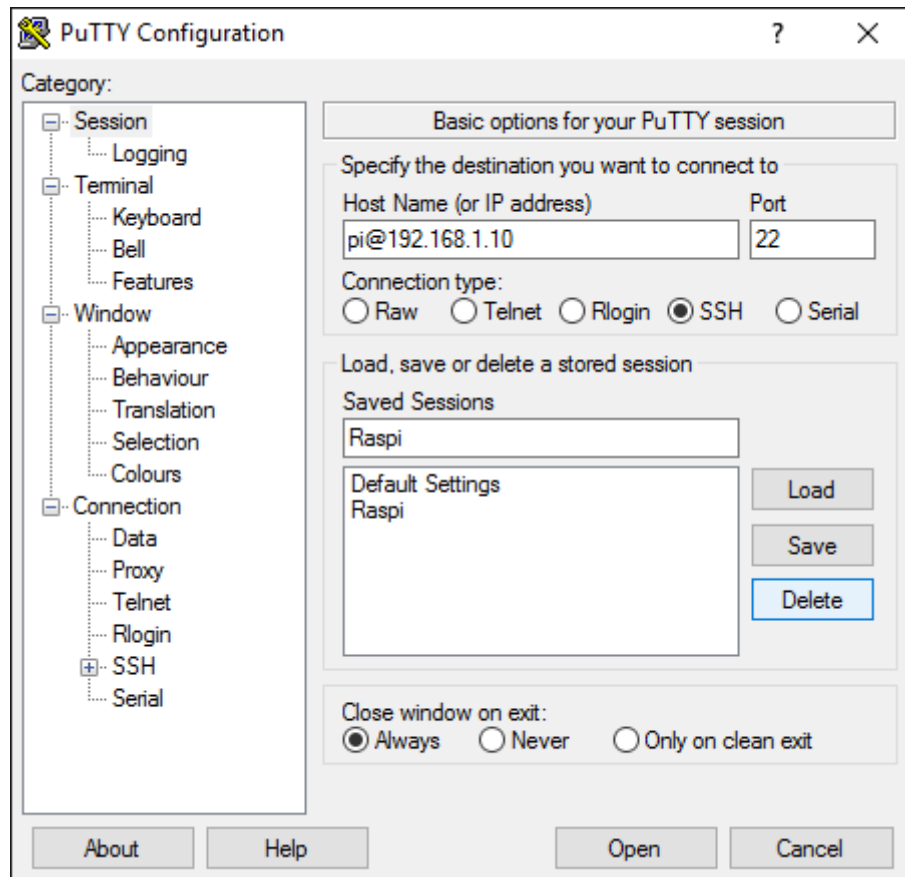
Ďalej bolo potrebné nastaviť časovú zónu, v ktorej sa nachádzame. Tieto údaje sú využívané pri nastavovaní hodín v RTC obvode. V prípade, že je Raspberry Pi pripojené k internetu, presný čas sa pri každom zapnutí aktualizuje z internetového servera.

V záložke *Advanced Options* je potrebné zapnúť automatické načítanie SPI a I²C pri spustení systému. Povolením prístupu k Raspberry Pi použitím SSH bol zabezpečený prístup k Raspberry Pi cez terminál.

Táto konfigurácia obsahuje ešte veľa nastavení, ale pre náš datalogger sú nepodstatné. Aby sa zmeny v nastaveniach uložili je potrebné zariadenie reštartovať.

Po zadaní príkazu `ifconfig` sme zistili priradenú IP adresu od smerovača, ktorá je potrebná pre pripojenie pomocou terminálu.

Jednou z možností terminálových programov je voľne dostupný program Putty. Po spustení do políčka *Host Name* sme napísali meno nášho zariadenia a IP adresu, ktorú má priradenú. Typ pripojenia sme zvolili SSH a port 22.



Obr. 6-3 Program Putty

Po pripojení stačí zadať len heslo, následne sa zariadenie pripojí na Raspberry Pi a je možné používať zariadenie cez vzdialený prístup.

Na koniec je ešte potrebné spraviť aktualizáciu systému. Aktualizáciu je vhodné urobiť aspoň raz za mesiac alebo po nainštalovaní nejakých nástrojov. Vykonáme ju nasledujúcimi príkazmi:

- `sudo apt-get update`
- `sudo apt-get upgrade [5]`.

6.1.2 Inštalácia nástrojov

Pre prácu s čipom BCM2835 v jazyku C bolo potrebné nainštalovať knižnicu k tomuto čipu. Datalogger využíva funkcie na obsluhu rozhrania SPI, zbernice I²C a GPIO portov. Túto voľne dostupnú knižnicu sme stiahli pomocou príkazu *sudo wget http://www.airspayce.com/mikem/bcm2835/bcm2835-1.46.tar.gz*. Postupným zadávaním nasledujúcich príkazov sme stiahnutý archív rozbalili, skompilovali a nainštalovali.

- `sudo tar zxvf bcm2835-1.46.tar.gz`
- `cd bcm2835-1.46`
- `sudo ./configure`
- `sudo make`

- `sudo make check`
- `sudo make install`

Ďalší balík, ktorý sme potrebovali, je podpora programovania v jazyku Python. Tento jazyk bol využívaný na zadávanie príkazov pre Raspberry Pi. Balík sme nainštalovali zadáním príkazov:

- `sudo apt-get install ipython3`
- `sudo apt-get install python-pip`
- `sudo apt-get install python3_rpi.gpio`

Pre prácu so zbernicou I²C, na ktorej máme pripojené hodiny reálneho času, bolo potrebné nainštalovať nástroje *python-smbus* a *i2c-tools*. Príkazom `sudo i2cdetect -y 0` môžeme overiť, či máme niečo na zbernici pripojené a na akej adrese.

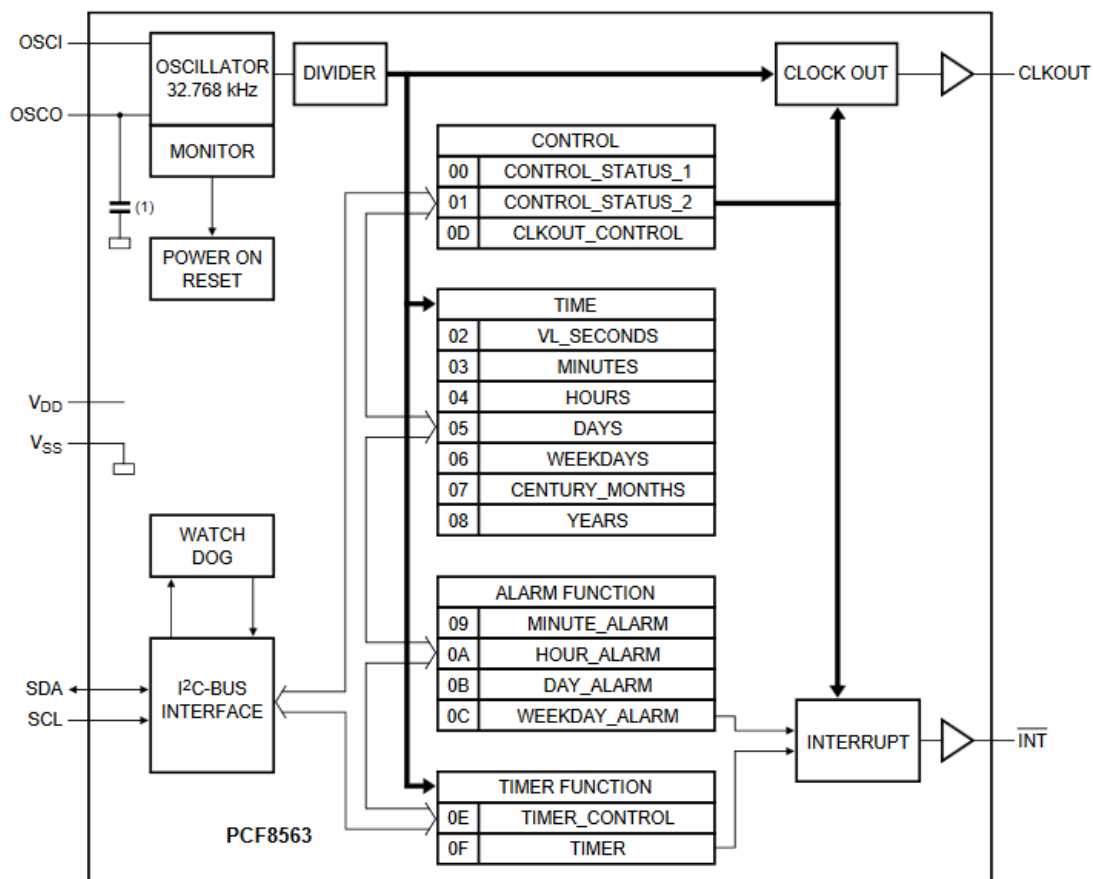
Na správu a ovládanie dataloggera používame webový server. Najčastejšie sa používa webový server Apache v kombinácii s Linuxom, MySQL a PHP, inak označovaný ako LAMP. Balíky sme nainštalovali zadáním nasledujúcich príkazov:

- `sudo apt-get install apache2 -y`
- `sudo apt-get install php5 libapache2-mod-php5 -y`
- `sudo apt-get install mysql-server php5-mysql -y`

Pri inštalácii nástroja MySQL sme zadali heslo do klienta „raspi“. Správne nainštalovanie si môžeme overiť zadáním IP adresy do webového prehliadača. Mala by sa zobraziť stránka uložená v Raspberry Pi `index.html`. [5][8]

6.2 Hodiny reálneho času

Zdroj hodín reálneho času je obvod PCF85633T. Pozostáva zo šestnástich 8 – bitových registrov. Tieto registre sú adresované od 0x00h až 0x0Fh. Prvé dva registre slúžia na zistenie stavu obvodu a nastavenie. Registre od adresy 0x02h až 0x08h sú určené pre čas a dátum. Nasledujúce registre po adresu 0x0Dh sa týkajú alarmu, ktoré datalogger nebude využívať. Posledné dva registre sú pre nastavenie časovača, sú dôležité, pretože od nich závisí perióda merania dát. Blokový diagram tohto obvodu môžeme vidieť na obr. 6-4. [13]



Obr. 6-4 Blokový diagram pcf8563 [13]

Formát registrov 0x02h až 0x08h môžeme vidieť v nasledujúcej tabuľke 6-1. Hodnoty v nich sú zapísané v BCD kóde.

Tab. 6-1 Registre 0x02 - 0x08 [13]

Adresa	Názov	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02h	Sekundy	x	0 – 59 sekunda						
0x03h	Minúty	x	0 – 59 minúta						
0x04h	Hodiny	x	x	0 – 23 hodina					
0x05h	Deň	x	x	1 – 31 deň					
0x06h	Deň v týždni	x	x	x	x	x	0 – 6 - > Ne - So		
0x07h	Mesiac	x	x	x	1 – 12 mesiac				
0x08h	Rok	0 – 99							

Obvod hodín reálneho času máme pripojený k Raspberry Pi cez zbernicu I²C. Maximálna rýchlosť komunikácie udaná výrobcom je 400 kHz. Pre tento datalogger až takú vysokú rýchlosť nepotrebujeme.

V jazyku C sme vytvorili knižnicu *pcf8563.h* pre tento obvod. Knižnica obsahuje niekoľko funkcií, ktoré sú potrebné pre datalogger. Na začiatku bolo potrebné určiť, akou rýchlosťou bude prebiehať komunikácia obvodu s Raspberry Pi. Zvolili sme 100 kHz.

Prvá funkcia `rtc_get_time`, ktorú využívame pri každom meraní dát má nasledovný zápis:

```
void rtc_get_time(char temp [7]){

    char addr [1] = {0x02};

    bcm2835_i2c_begin();
    bcm2835_i2c_setSlaveAddress(RTC_ADDR);
    bcm2835_i2c_setClockDivider(BCM2835_I2C_CLOCK_DIVIDER_2500;
    bcm2835_i2c_read_register_rs(addr, temp, 7);
    sleep(0.001);
    bcm2835_i2c_end();

    temp[1] &= 0x7f;
    temp[2] &= 0x3f;
    temp[3] &= 0x3f;
    temp[4] &= 0x07;
    temp[5] &= 0x9f;
}
```

Táto funkcia načíta dátum a čas do poľa *temp* cez zbernicu I²C od adresy registra 0x02h. Počet čítaných dát je 7 bytov, čo odpovedá kompletnému údaju o čase a dátume. V poli sú postupne usporiadané hodnoty tak, ako nasledujú registre za sebou. Pre adresu obvodu hodín reálneho času na zbernici máme zadané makro:

```
#define RTC_ADDR 0x51
```

Veľmi podobná funkcia je `rtc_set_time`. Líši sa od predchádzajúcej len tým, že hodnoty z poľa *temp* zapíše do registrov obvodu.

Ďalšiu využívanú funkciu je `set_time_to_RTC`.

```
void set_time_to_RTC(void){

    struct tm *sys_time;
    struct timeval curTime;
    char time[7] = {0x0};
```

```

//získanie času z RPI
    gettimeofday(&curTime, NULL);
    sys_time = localtime(&curTime.tv_sec);
    sys_time->tm_year -= 100;
    sys_time->tm_mon += 1;
//prepočet času tm do bcd
    time[6] = itobcd(sys_time->tm_year);
    time[5] = itobcd(sys_time->tm_mon);
    time[4] = itobcd(sys_time->tm_wday);
    time[3] = itobcd(sys_time->tm_mday);
    time[2] = itobcd(sys_time->tm_hour);
    time[1] = itobcd(sys_time->tm_min);
    time[0] = itobcd(sys_time->tm_sec);

    rtc_set_time(time);

//nastavenie hodín v RTC
}

```

Táto funkcia pracuje z knižnicou *time.h*, z ktorej používame štruktúry a funkcie na získanie časového údajja zo systému. Časový údaj je následne prevedený do formátu vhodného na zápis do obvodu RTC.

Períodu merania zabezpečuje časovač, ktorého dva registre sa nachádzajú od adresy 0x0Eh. V prvom registri je potrebné nastaviť bit 7, čím povolíme časovač. Kombináciou bitov 0 a 1 môžeme nastaviť frekvenciu počítania časovača. Nastavili sme frekvenciu 1 Hz. Do druhého registra sa zapíše počet sekúnd, po ktorých má časovač vykonať prerušenie na pine INT.

```

void timer_init(uint8_t second){

    char temp [2] = {0x01, 0x11};
    char addr [1] = {0x0E};

    bcm2835_i2c_begin();
    bcm2835_i2c_setSlaveAddress(RTC_ADDR);
    bcm2835_i2c_setClockDivider(BCM2835_I2C_CLOCK_DIV_2500);
    bcm2835_i2c_write(temp, 2);
//Povolenie prerušenia

    temp [0] = 0x0f;
    temp [1] = second;
    bcm2835_i2c_write(temp, 2);
}

```

```

//nastavenie sekúnd

temp [1] = 0x02;
temp [0] = addr[0];
bcm2835_i2c_write(temp, 2);
//Nastavenie zdroja pre časovač

bcm2835_i2c_end();
}

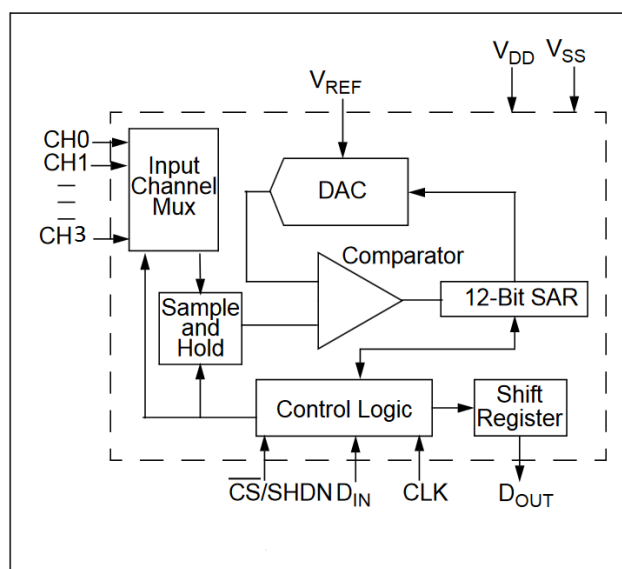
```

Funkciami `timer_start` a `timer_stop` je možné časovač ovládať. To vykonáme zapísaním príslušných hodnôt do registra 0x0Eh.

Knižnica obsahuje ďalšie funkcie, ako napríklad `read_period`, ktorá načítava periódu merania z databázy.

6.3 Analógovo-digitálny prevodník MCP3204

Datalogger zaznamenáva dva napäťové vstupy, jeden prúdový a stav napätia batérie. Tieto analógové napätia sme previedli na číslo pomocou AD prevodníka MCP3204 od firmy Microchip. Jedná sa o 12-bitový prevodník s postupnou aproximáciou [11]. Obsahuje štyri nezávislé vstupy alebo dva diferenciálne. Použili sme formu štvorice nezávislých vstupov. Na vstup CH0 je pripojené upravené napätie pre analógový vstup 0 – 5 V, na vstup CH1 pre analógový vstup 0 – 10 V, na CH2 prevedený prúd na napätie a na vstup CH3 napätie na záložnej batérii. Tieto napätia sú porovnávané s referenčným napätím 4,096 V a vyhodnotené na číslo. Blokovú schému tohto prevodníka môžeme vidieť na obr. 6-5.



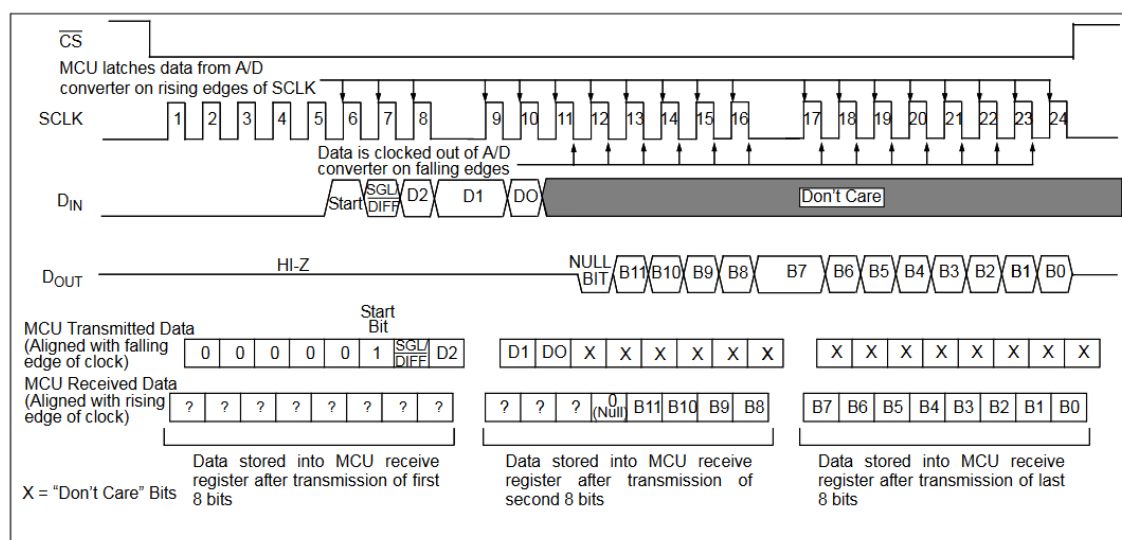
Obr. 6-5 Blokový diagram AD prevodníka[11]

Výsledné napätie vypočítame z rovnice:

$$U_{OUT} = \frac{U_{REF} \cdot D}{2^{12}} [V] \quad (6.1)$$

U_{ref} predstavuje hodnotu referenčného napätia, v našom prípade 4,096 V, D predstavuje číslo, na ktoré prevodník prevedie vstupné analógové napätie.

Prevodník je vybavený rozhraním SPI, ktorým je pripojený cez galvanické oddelenie na Raspberry Pi signálmi SPI_MOSI, SPI_MISO, SPI_CLK a SPI_CS. Maximálna rýchlosť prevodníka je 100 kHz. Raspberry Pi má základnú frekvenciu 250 MHz. Pri použití deličky 16384 dostaneme výslednú frekvenciu 15,26 kHz. Táto frekvencia je pre datalogger dostačujúca. Na obr. 6-6 môžeme vidieť princíp komunikácie AD prevodníka cez SPI.



Obr. 6-6 Princíp komunikácie prevodníka[11]

Z obrázku vyplýva, že na získanie prevedeného napätia je potrebné poslať po SPI signálom MOSI tri bajty dát. Prvé dva bajty obsahujú údaje o kanáli, ktorého vstupné napätie má byť prevedené na číslo. Na hodnote tretieho bajtu nezáleží. Musí sa posilať len preto, aby bolo dokončené vysielanie prevodníka kvôli úplnej hodnote čísla odpovedajúcemu prevedeného napätia. Toto číslo Raspberry Pi prijme po signáli MISO. Hodnota prvého bajtu je pre nás nepodstatná. Výsledné prevedené číslo dostaneme ako spodné štyri bity druhého prijatého bajtu a celý tretí bajt.

Podľa tab. 6-2 zistíme, aké hodnoty musia obsahovať prvé dva bajty odoslané z Raspberry Pi. Keďže nepoužívame diferenciálne zapojenie kanálov prevodníka, bit Single/Diff má hodnotu 1. Bit D2 môže mať obe hodnoty. Zvolili sme hodnotu 0. Kombinácia bitov D1 a D0 určuje, ktorý kanál bude prevádzaný. Prvý vysielaný bajt z Raspberry Pi je pre všetky štyri kanály rovnaký s hodnotou 0x06h. Druhý vysielaný

bajt je pre každý kanál iný. V poradí od kanálu 0 sú hodnoty bajtu 0x00h, 0x40h, 0x80h a 0xC0h.

Tab. 6-2 Riadiace bity[11]

Riadiace bity				Výber kanála
Single/Diff	D2	D1	D0	
1	x	0	0	CH 0
1	x	0	1	CH 1
1	x	1	0	CH 2
1	x	1	1	CH 3
0	x	0	0	CH 0 = IN+ CH 1 = IN-
0	x	0	1	CH 0 = IN- CH 1 = IN+
0	x	1	0	CH 2 = IN+ CH 3 = IN-
0	x	1	1	CH 2 = IN- CH 3 = IN+

Na obsluhu tohto AD prevodníka sme vytvorili knižnicu *datalogger.h*. Obsahuje funkciu `read_analog_inputs`, ktorej má zápis:

```
void read_analog_inputs(float res[4]){

    char rbuf[3] = {0x0};
    char tbuf[3] = {0x06, 0x00, 0x00};
    char tbuf2[4] = {0x00, 0x40, 0x80, 0xc0};
    uint32_t len = 3;
    uint16_t temp [4] = {0};
    int i;

    bcm2835_spi_begin();
    bcm2835_spi_setBitOrder(BCM2835_SPI_BIT_ORDER_MSBFIRST);
    bcm2835_spi_setDataMode(BCM2835_SPI_MODE0);

    bcm2835_spi_setClockDivider(BCM2835_SPI_CLOCK_DIVIDER_16384);
    bcm2835_spi_setChipSelectPolarity(BCM2835_SPI_CS0, LOW);
    bcm2835_spi_chipSelect(BCM2835_SPI_CS0);
    for(i=0;i<4;i++){
        tbuf[1] = tbuf2[i];
        bcm2835_spi_transfernb(tbuf,rbuf,len);
    }
}
```

```

        temp[i] = (rbuf[1] << 8) | rbuf[2];
        res[i] = (float)temp[i]*4.11/4096;
    }
    res[0] = res[0]*2;
    res[1] = res[1]*4;
    res[2] = res[2]*1000/115;
//    prepočet čísel na napätie
    if((res[0]>6) | (res[0]<-1))
        res[0] = 0;
    if((res[1]>11) | (res[1]<-1))
        res[1] = 0;
    if((res[2]>33) | (res[2]<-1))
        res[2] = 0;
//    nastavenie hraníc hodnôt
    bcm2835_spi_end();
}

```

Na začiatku tela tejto funkcie máme zadané polia, ktoré slúžia ako zásobníky na odosielanie a prijímanie dát. Ďalej je potrebné nastaviť samotnú komunikáciu po SPI a to parametre ako MSB/LSB prvý bit, rýchlosť, SPI mód, polaritu CS (ChipSelect) a samotný CS, ktorý máme pripojený.

Prijaté dáta sa ďalej prepočítajú na výsledné namerané napätia a v prípade kanálu 2 na prúd prechádzajúci odporom. Výsledky sa uložia do premennej float `res [4]`.

6.4 Čítanie digitálnych vstupov

Funkcia `read_digital_inputs` sa tiež nachádza v knižnici *datalogger.h*. Pozostáva z nastavení digitálnych vstupov na GPIO porty. Všetky porty sú nastavené ako vstupné a je im priradený pull – up rezistor. Namerané hodnoty sú uložené do premennej `uint8_t data [8]`.

```

void read_digital_inputs(uint8_t data[8]){

    bcm2835_gpio_fsel(DI0, INPUT);
    bcm2835_gpio_fsel(DI1, INPUT);
    bcm2835_gpio_fsel(DI2, INPUT);
    bcm2835_gpio_fsel(DI3, INPUT);
    bcm2835_gpio_fsel(DI4, INPUT);
    bcm2835_gpio_fsel(DI5, INPUT);
    bcm2835_gpio_fsel(DI6, INPUT);
    bcm2835_gpio_fsel(DI7, INPUT);
//    nastavenie gpio ako input

```



```

bcm2835_gpio_set_pud(DI0, PULL_UP);
bcm2835_gpio_set_pud(DI1, PULL_UP);
bcm2835_gpio_set_pud(DI2, PULL_UP);
bcm2835_gpio_set_pud(DI3, PULL_UP);
bcm2835_gpio_set_pud(DI4, PULL_UP);
bcm2835_gpio_set_pud(DI5, PULL_UP);
bcm2835_gpio_set_pud(DI6, PULL_UP);
bcm2835_gpio_set_pud(DI7, PULL_UP);

//      nastavenie pull-up

data[0] = (!bcm2835_gpio_lev(DI0));
data[1] = (!bcm2835_gpio_lev(DI1));
data[2] = (!bcm2835_gpio_lev(DI2));
data[3] = (!bcm2835_gpio_lev(DI3));
data[4] = (!bcm2835_gpio_lev(DI4));
data[5] = (!bcm2835_gpio_lev(DI5));
data[6] = (!bcm2835_gpio_lev(DI6));
data[7] = (!bcm2835_gpio_lev(DI7));

//      čítanie vstupov
}

```

6.5 Databáza

Namerané dáta dataloggeru je potrebné niekam ukladať pre potreby sa späť dohľadieť k týmto dátam. Jedným zo spôsobov ako ukladať dáta je napríklad zápis do textového súboru. Ďalším spôsobom je databázový systém a tento spôsob sme vybrali aj pre tento datalogger. Používame databázový systém MySQL, ktorý po doinštalovaní podporuje aj Raspberry Pi. Ide o často používaný systém pre stredné veľkosti databáz, ktorý je vhodný pre prácu s webovými aplikáciami [18].

Najskôr sme vytvorili databázu cez klienta MySQL, ktorého sme pridali pri inštalácii Raspberry Pi. Spustili sme ho zadaním príkazu: `mysql -u root -praspri`.

„Root“ predstavuje prvotného užívateľa a „raspi“ heslo, ktoré sme zadali pri inštalácii klienta. Po spustení klienta sme vytvorili vlastného užívateľa pre datalogger. Užívateľ sa volá „datalogger“ a nastavili sme mu heslo „tomas6“.

```

- CREATE USER 'datalogger'@'localhost' IDENTIFIED BY
  'tomas6';

```

Ak bol užívateľ správne vytvorený, klient nám vypíše *QUERY* → *OK*. Táto odpoveď sa bude zobrazovať po každom vykonanom príkaze. Inak napíše chybu.

Ďalej sme vytvorili databázu *temp*, s ktorou budeme pracovať a ukladať do nej dáta.

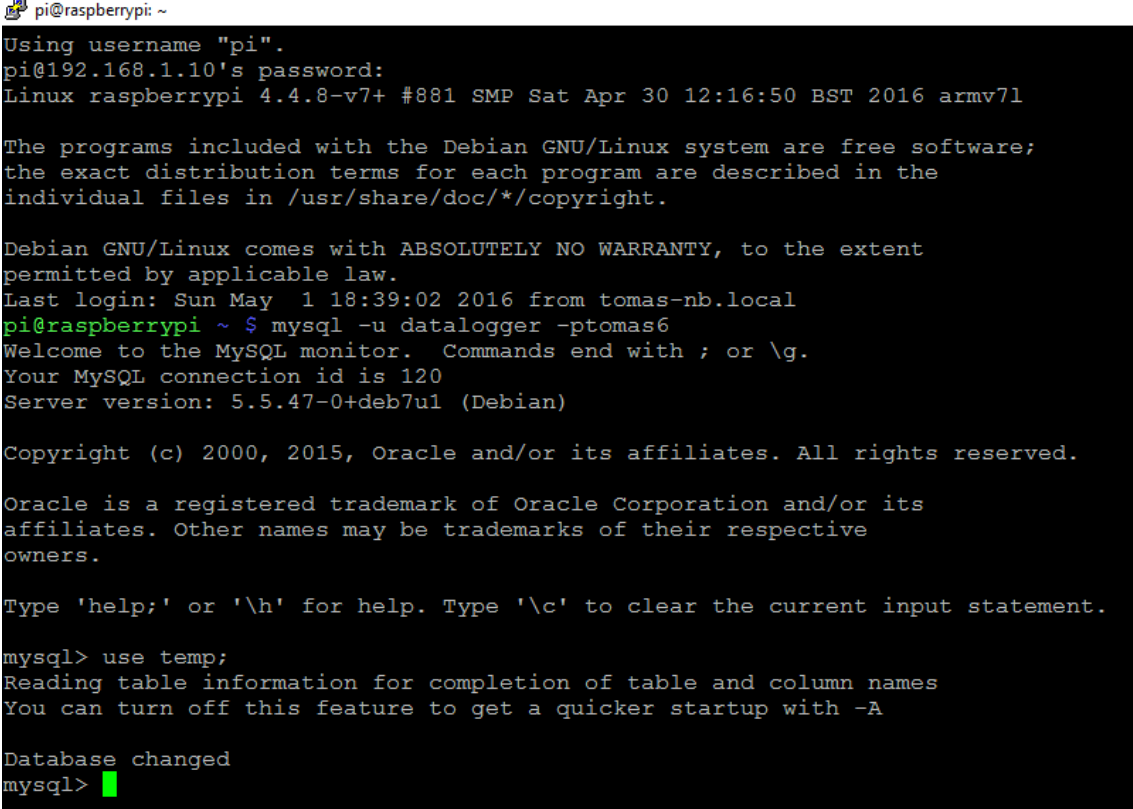
- `CREATE DATABASE temp;`

Vytvorenému užívateľovi sme prideliť všetky práva pre prácu s novo vytvorenou databázou.

- `GRANT ALL PRIVILEGES ON temp.* TO 'datalogger'@'localhost';`
- `FLUSH PRIVILEGES;`
- `quit;`

Po pridelení práv novému užívateľovi sme opustili klienta. Následne sme sa prihlásili už cez užívateľa „datalogger“ s heslom „tomas6“. Ďalším príkazom sme zabezpečili prácu s databázou *temp*.

- `mysql -u datalogger -ptomas6`
- `use temp;`



```
pi@raspberrypi: ~
Using username "pi".
pi@192.168.1.10's password:
Linux raspberrypi 4.4.8-v7+ #881 SMP Sat Apr 30 12:16:50 BST 2016 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun May 1 18:39:02 2016 from tomas-nb.local
pi@raspberrypi ~ $ mysql -u datalogger -ptomas6
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 120
Server version: 5.5.47-0+deb7u1 (Debian)

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use temp;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql>
```

Obr. 6-7 MySQL klient

Namerané dáta sú ukladané do tabuliek. Vytvorili sme hlavičku tabuľky pre analógové vstupy pomocou príkazu:

- `CREATE TABLE analog_inputs(ID int NOT NULL AUTO_INCREMENT, RTC_DATE date, RTC_TIME time, 0_5V float(7,3), 0_10V float(7,3), 20mA float(7,3), RTC_BAT float(7,3), TIME_TAG timestamp, PRIMARY KEY(ID));`

Vytvorili sme si tabuľku (obr. 6-8) s ôsmimi stĺpcami. Každý stĺpec má svoj názov a za ním typ hodnoty, ktoré sú doňho zapisované. Stĺpec *ID* slúžiť ako nenulový index pre každý zapísaný riadok iný. Podľa tohto indexu môžeme v tabuľke vyhľadávať bez zadania ďalších informácií o zázname. Vytvorenú tabuľku si pozrieme zadáním príkazu:

- `SHOW COLUMNS FROM analog_inputs;`

```
mysql> SHOW COLUMNS FROM analog_inputs;
```

Field	Type	Null	Key	Default	Extra
ID	int(11)	NO	PRI	NULL	auto_increment
RTC_DATE	date	YES		NULL	
RTC_TIME	time	YES		NULL	
0_5V	float(7,3)	YES		NULL	
0_10V	float(7,3)	YES		NULL	
20mA	float(7,3)	YES		NULL	
RTC_BAT	float(7,3)	YES		NULL	
TIME_TAG	timestamp	NO		CURRENT_TIMESTAMP	on update CURRENT_TIMESTAMP

8 rows in set (0.01 sec)

```
mysql>
```

Obr. 6-8 Vytvorená tabuľka

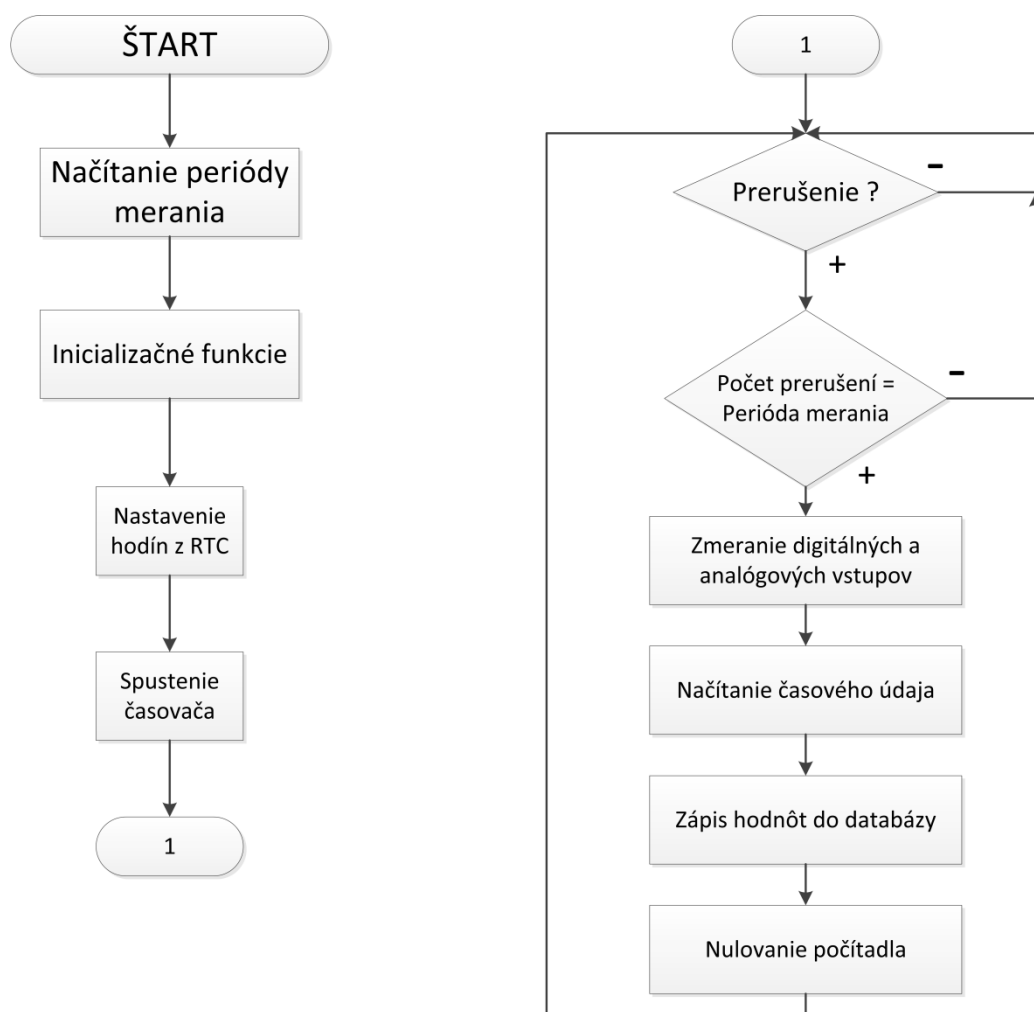
Ďalšie tabuľky, ktoré sme potrebovali vytvoriť, sú *digital_inputs* a *period*. Prvá z nich je podobná predchádzajúcej tabuľke, ale s digitálnymi vstupmi. Posledná tabuľka je používaná na uloženie hodnoty periódy merania dataloggera.

- `CREATE TABLE digital_inputs(ID int NOT NULL AUTO_INCREMENT, RTC_DATE date, RTC_TIME time, DI_0 int(1), DI_1 int(1), DI_2 int(1), DI_3 int(1), DI_4 int(1), DI_5 int(1), DI_6 int(1), DI_7 int(1), TIME_TAG timestamp, PRIMARY KEY(ID));`
- `CREATE TABLE period(ID int NOT NULL AUTO_INCREMENT, PERIOD int(32), PRIMARY KEY(ID));`

S vytvorenými tabuľkami sme ďalej pracovali v hlavom programe dataloggera v jazyku C a na webových stránkach v skriptovacom jazyku PHP. Inštrukcie ako pracovať z databázovým systémom MySQL sme čerpali z internetového zdroja [15].

6.6 Hlavná slučka

Celá činnosť dataloggeru je založená na hlavnom programe vytvorenom v module *main.c*. Zahŕňa v sebe funkcie obsluhujúce meranie dát na vstupoch a funkcie pre prácu z databázovým systémom MySQL. Vývojový diagram princípu dataloggeru môžeme vidieť na obr 6-9. Program používa okrem našich vytvorených knižníc *datalogger.h* a *pcf8563.h*, aj knižnicu *bcm2835.h*, ktorá obsahuje funkcie na používanie periférií Raspberry Pi a knižnicu *mysql.h* potrebnú k práci s databázou.



Obr. 6-9 Vývojový diagram

Na začiatku máme zadané makrá, ktoré používame na prihlásenie do databázy.

```
#define DATABASE_NAME "temp"
#define DATABASE_USERNAME "datalogger"
```

```
#define DATABASE_PASSWORD "tomas6"
```

Na začiatku programu sme zavolali funkciu `read_period`, ktorá nám načítala hodnotu periódy merania v sekundách uloženú v databázovej tabuľke. Zápis funkcie je:

```
uint32_t read_period(void){

    #define DATABASE_NAME "temp"
    #define DATABASE_USERNAME "datalogger"
    #define DATABASE_PASSWORD "tomas6"

    MYSQL *mysql1;
    MYSQL_RES *res;
    MYSQL_ROW row;

    char str[120] = "SELECT PERIOD FROM period WHERE ID=1";
    uint32_t sec = 0;

    mysql1 = mysql_init(NULL);
    sec = 0;

    if(mysql_real_connect(mysql1, "localhost",
        DATABASE_USERNAME,DATABASE_PASSWORD,
        DATABASE_NAME, 0,NULL,0) == NULL)
        printf("Connect failed!!\n");

    //pripojenie do databazy
    mysql_query(mysql1, str);
    res = mysql_use_result(mysql1);
    //nacitanie udaja z databazy
    row = mysql_fetch_row(res);
    sec = atoi(row[0]);
    mysql_free_result(res);
    mysql_close(mysql1);
    return(sec);
}
```

Následne sa vykonalo niekoľko inicializačných funkcií. Funkcia `bcm2835_init` nastavila obvod `bcm2835` na prácu s perifériami a GPIO portami. Ďalšou funkciou `set_time_from_RTC` boli nastavené hodiny v Raspberry Pi načítané z obvodu hodín reálneho času PCF8563. Ďalej sa zinicilizoval časovač funkciou `timer_init(1)`, kde vstupný parameter je počet sekúnd, ako často má byť vyvolané prerušenie. Časovač

sme následne spustili funkciou `timer_start`. Program ďalej pokračuje v nekonečnej slučke `while(1)`. V každom cykle program čaká na vzostupnú hranu prerušenia od hodín reálneho času. Keď príde nastavený počet prerušení začína proces merania a zapisovania dát. Najskôr sa rozsvieti LED dióda *LED1* funkciou `bcm2835_gpio_write(LED1, LOW)`. Rozsvietená LED dióda nám signalizuje, že datalogger sa nachádza v stave merania. Následne sa pripojíme do databázy, zmeriame vstupy funkciami `read_digital_inputs` a `read_analog_inputs`, a funkciou `rtc_get_time` načítame údaj o čase, kedy boli hodnoty namerané.

```
read_digital_inputs(d_inputs);
read_analog_inputs(a_inputs);
rtc_get_time(time);
//zmeranie hodnôt
sprintf(digital, "INSERT INTO digital_inputs (RTC_DATE,
        RTC_TIME, DI_0, DI_1, DI_2, DI_3, DI_4, DI_5, DI_6,
        DI_7) VALUES ('20%02x-%02x-%02x', '%02x:%02x:%02x',
        %d, %d, %d, %d, %d, %d, %d, %d)", time[6], time[5],
        time[3], time[2], time[1], time[0], d_inputs[0],
        d_inputs[1], d_inputs[2], d_inputs[3], d_inputs[4],
        d_inputs[5], d_inputs[6], d_inputs[7]);

sprintf(analog, "INSERT INTO analog_inputs (RTC_DATE,
        RTC_TIME, 0_5V, 0_10V, 20mA, RTC_BAT)
        VALUES ('20%02x-%02x-%02x', '%02x:%02x:%02x', %2.4f,
        %2.4f, %2.4f, %2.4f)", time[6], time[5], time[3],
        time[2], time[1], time[0], a_inputs[0], a_inputs[1],
        a_inputs[2], a_inputs[3]);
//vytvorenie príkazov do mysql
mysql_query(mysql1, digital);
mysql_query(mysql1, analog);
//zapísanie do databázy
mysql_close(mysql1);
bcm2835_gpio_write(LED1, HIGH);
```

Po zapísaní nameraných údajov do tabuliek databázy sa ukončí spojenie s databázou a zhasne LED dióda. Celý proces sa opakuje, keď počet prerušení dosiahne nami požadovanú hodnotu.

Na kompiláciu programu bol použitý kompilátor GCC. K štandardnému príkazu na kompilovanie bolo potrebné pridať nami vytvorené knižnice a knižnice pre prácu s MYSQL a s čipom BCM2835. Tento príkaz má tvar:

```
- sudo gcc -Wall -o main main.c libpcf8563.a libdatalogger.a  
-lbcm2835 -lmysqlclient
```

Hlavný program je možné spúšťať z webovej stránky v záložke nastavenia.

6.7 Webová stránka

Súčasťou návrhu dataloggeru bola administrácia nameraných údajov prostredníctvom webového rozhrania. Jeden zo spôsobov administrácie týchto údajov je použitie webového serveru Apache, ktorý nám tieto stránky poskytuje. Webový server Apache sa automaticky spúšťa po nabenutí systému v Raspberry Pi. Stránky, ktoré sa zobrazujú na lokálnej IP adrese pridelennej pre Raspberry Pi na porte 80, sú napísané kombináciou značkovacieho jazyka HTML a skriptovacích jazykov PHP a JavaScript. Postup ako navrhovať webové stránky sme čerpali z internetového zdroja [15].

Všetky stránky sú uložené v pamäťovej karte Raspberry Pi v adresári *var/www*. Vytvorili sme súbor *mystyle.css*, ktorý efektívne definuje štýl tabuľky, menu, celkový vzhľad stránky, pozadie a rozmery jednotlivých častí pre každú našu stránku je možné načítať ho v hlavičke každej stránky nasledujúcim zápisom:

```
<link rel="stylesheet" type="text/css" href="mystyle.css">
```

Ako prvá sa nám zobrazí stránka *index.php*. Rozdiel medzi použitím a nepoužitím súboru so štýlmi môžeme vidieť na nasledujúcich dvoch obrázkoch.



Datalogger s ethernet rozhraním

- [Úvod](#)
- [Zmerané dáta](#)
- [Nastavenia](#)
- [Informácie](#)



© Tomáš Orávik - Diplomová práca 2016 email: xoravi00@stud.feec.vutbr.cz

Obr. 6-10 Webová stránka nedefinovaná súborom *mystyle.css*



Datalogger s ethernet rozhraním

Úvod
Zmerané dáta
Nastavenia
Informácie

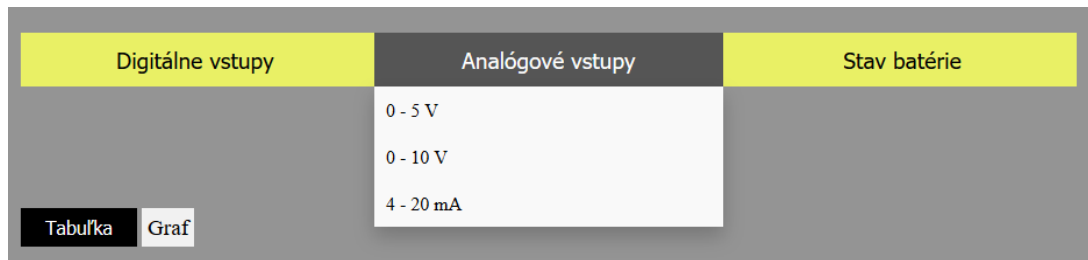


Obr. 6-11 Webová stránka definovaná súborom *mystyle.css*

Obsah stránok je rozložený do štyroch blokov. V hornej časti sa nachádza blok, v ktorom je logo školy a hlavný nadpis. V spodnej časti sa nachádza blok, ktorý je pre všetky stránky rovnaký a zobrazujú sa v ňom kontaktné údaje autora. Blok v strednej časti webových stránok je dynamický, na každej stránke zobrazuje niečo iné. Naľavo sa nachádza blok hlavného menu, ktorý pozostáva zo štyroch záložiek. V záložke *Úvod* sa zobrazuje fotografia nášho datalogeru, záložka *Zmerané dáta* slúži na zobrazenie nameraných hodnôt vo forme tabuľky alebo grafu, v záložke *Nastavenia* je možné nastavenie dataloggeru po zadaní hesla sprístupnia (6.8) a posledná záložka obsahuje základné informácie o našom dataloggeri.

V záložke *Zmerané dáta* sme vytvorili dvojúrovňové rolovacie menu, so základným rozdelením na tri skupiny a to *Analógové vstupy*, *Digitálne vstupy* a *Stav batérie*. Pod každou skupinou sa nachádzajú príslušné vstupy. Zápis tohto rolovacieho menu pre skupinu analógových vstupov v kóde HTML je nasledovný:

```
<div class="dropdown" style="right:0">  
  <button class="dropbtn">Analógové vstupy</button>  
  <div class="dropdown-content" style="right:0">  
    <a href="data_5V_tab.php">0 - 5 V</a>  
    <a href="data_10V_tab.php">0 - 10 V</a>  
    <a href="data_20mA_tab.php">4 - 20 mA</a>  
  </div>  
</div>
```

Obr. 6-12 Rolovacie menu

Pre lepšie hľadanie nameraných hodnôt jednotlivých vstupov, sú pre každý vstup vytvorené tabuľky a grafy. Tabuľky sú zložené zo štyroch stĺpcov. Stĺpec *ID* nám zobrazuje indexové číslo nameranej hodnoty. Ďalšie dva stĺpce nám zobrazujú časový údaj, kedy bola hodnota zmeraná. Posledný stĺpec udáva nameranú hodnotu.

Tabuľky sú generované pomocou PHP skriptu. Najskôr je potrebné prihlásiť sa do databázy a vybrať z nej údaje. Tieto údaje sú potom postupne zapisované vo *while* cykle do tabuľky až dokiaľ sa nezapíšu všetky riadky. Pre vypísanie tabuľky stavu RTC batérie má PHP skript zápis:

```
<?php
if ($result->num_rows > 0){
    echo"<div class='scroll'><table><tr><th>ID</th>
        <th>Dátum</th><th>Čas</th>
        <th>Hodnota [V]</th></tr>";
    //vytvorenie hlavičky pre tabuľku

    while($row = $result->fetch_assoc()){
        echo "<tr><td>" . $row["ID"]. "</td><td>" .
            row["RTC_DATE"].
            "</td><td>".$row["RTC_TIME"]."</td><td>".
            $row["RTC_BAT"]. "</td></tr>";
        //zapísanie hodnôt do tabuľky
    } else {
        echo "</table></div>";
    } else {
        echo "0 results";
    }?>
```

Vytvorenú tabuľku s nameranými hodnotami môžeme vidieť na nasledujúcom obrázku (obr. 6-13).

Stav RTC batérie			
Tabuľka		Graf	
ID	Dátum	Čas	Hodnota [V]
89	2016-05-05	22:34:15	0.872
88	2016-05-05	22:32:15	0.875
87	2016-05-05	22:30:15	0.874
86	2016-05-05	22:28:15	0.873
85	2016-05-05	22:26:15	0.872
84	2016-05-05	22:24:15	0.872
83	2016-05-05	22:22:15	0.872
82	2016-05-05	22:20:15	0.873
81	2016-05-05	22:18:15	0.872
80	2016-05-05	22:16:15	0.871
79	2016-05-05	22:14:15	0.872
78	2016-05-05	22:12:15	0.872
77	2016-05-05	22:10:15	0.870
76	2016-05-05	22:08:15	0.870

Obr. 6-13 Vytvorená tabuľka v PHP

Na grafické zobrazenie nameraných hodnôt nám už nestačí len PHP jazyk, pretože patrí medzi dynamické objekty. Najčastejšie používaný jazyk vhodný na vykresľovanie grafov na webových stránkach je JavaScript. Na vytvorenie grafu sme použili voľne dostupné knižnice pre nekomerčné účely stiahnuté z internetu *highcharts.js* a *highstock.js* [19]. Jazyk JavaScript nepodporuje prácu z databázami, preto ho používame v kombinácii s jazykom PHP.

V hlavičke súboru *data_d3_graf.php* sme si vytvorili funkciu *container.highcharts*, v ktorej sú nadefinované všetky vlastnosti a nastavenia grafu, ktorý je vykresľovaný. Patria sem názvy grafov a osí, jednotky osí, typ grafu, farba charakteristiky, legenda a iné. Dáta, ktoré majú byť vykresľované sú uložené do polí. Z poľa *data_time* sa načítavajú hodnoty pre os X a z poľa *data_value* hodnoty pre os Y pomocou PHP skriptu. Najskôr je potrebné sa spojiť s databázou a načítať v tabuľke hodnoty stĺpca *DI_3* a *TIME_TAG*.

<?php

```
$servername = "localhost";
$username = "datalogger";
$password = "tomas6";
$dbname = "temp";
```

```

        //Vytvorenie spojenia
$conn = new mysqli($servername, $username, $password,
                    $dbname);
//Overenie spojenia
if ($conn->connect_error){
    die("Connection failed: ".$conn->connect_error);
}
//Načítanie údajov z tabuľky
$sql = "SELECT DI_3, TIME_TAG, RTC_DATE, RTC_TIME
        from digital_inputs ORDER BY TIME_TAG ASC";
$result = $conn->query($sql);
?>

```

Následne boli načítané hodnoty vložené do poľa *data_time*.

```

<script>
    //Naplnenie poľa časovou značkou
    var data_time = [<?php
        while ($row = mysqli_fetch_array($result))
            echo "'".$row['RTC_DATE']."'."'.$row['RTC_TIME']."'";
    ?>];
</script>

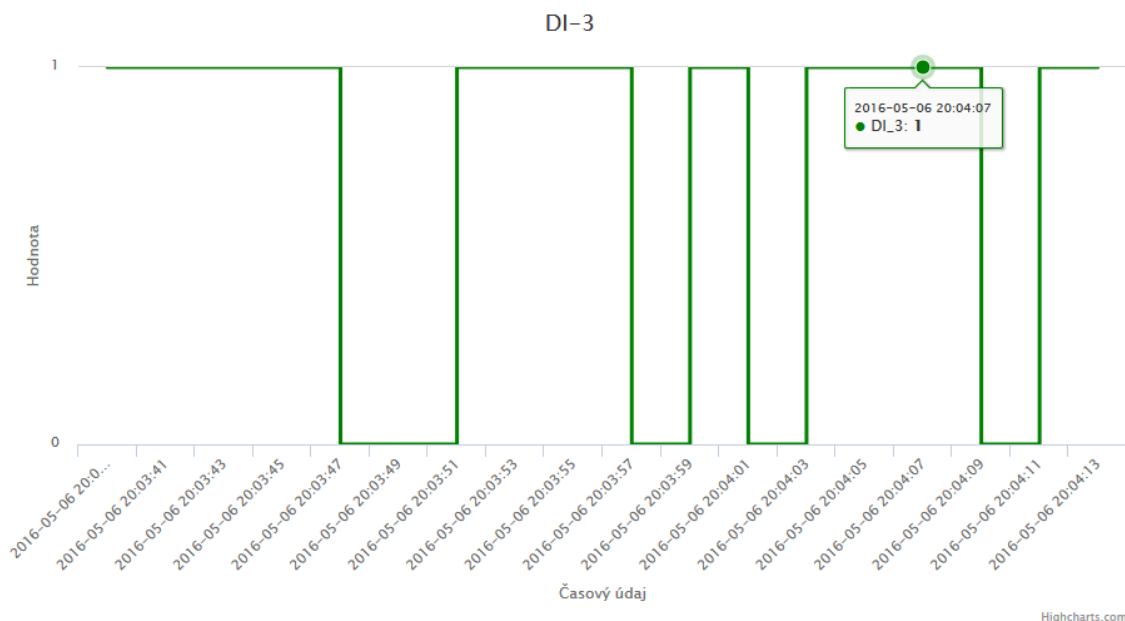
```

Ďalej musia byť znova načítané a následne zapísané do poľa *data_value* hodnoty z databázovej tabuľky, pretože funkcia *mysqli_fetch_array* postupne načítané hodnoty vymazáva.

```

<?php
    //Načítanie údajov z tabuľky
    $sql = "SELECT DI_3, TIME_TAG from digital_inputs
            ORDER BY TIME_TAG ASC";
    $result = $conn->query($sql);
    ?>
<script>
    //Naplnenie poľa hodnotou DI_3
    var data_value = [<?php
        while ($row = mysqli_fetch_array($result))
            echo $row['DI_3'].',';
    ?>];
</script>

```



Obr. 6-14 Grafické zobrazenie nameraných hodnôt

Pre každý vstup dataloggeru sú napísané dve webové stránky. Jedna pre tabuľku a druhá pre graf. Poslednou záložkou hlavného menu *Nastavenia* sa podrobne zaoberáme v nasledujúcej podkapitole.

6.8 Ovládanie dataloggeru

Koncept dataloggeru je navrhnutý tak, aby bolo ovládanie a všetky nastavenia vykonávané pomocou webového rozhrania. V záložke *Nastavenia* sa po zadaní prihlasovacieho hesla "admin" otvorí stránka *settings2.php*.

Obr. 6-15 Nastavenia dataloggeru

Forma nastavení je vytvorená pomocou menu, ktoré tvoria zaškrŕavacie políčka a tlačidlo *Použiť*. Každá možnosť má svoju hodnotu, ktorú stlačením tlačidla preniesieme do PHP skriptu *status.php*. V skripte sa vyhodnotí, o ktorú možnosť sa jedná a následne sa vykonajú potrebné nastavenia.

Možnosti *Spustiť meranie*, *Zastaviť meranie*, *Vypnúť* a *Reštart* sú podobné. Pri vykonávaní týchto možností sa cez jednoduchý PHP skript spustí program napísaný v jazyku Python [20]. Tento program pomocou funkcie *os.system* zadá príkaz do Raspberry Pi, ako keby bol príkaz napísaný do terminálu. Zápis PHP skriptu a programu *log_on.py*:

```
<?php
$stat = $_POST['status'];

if($stat == 'log_on'){
    exec("sudo python /home/pi/datalogger/python/log_on.py");
    header("location:settings.php");
}

if($stat == 'log_off'){
    header("location:settings2.php");
    exec("sudo python /home/pi/datalogger/python/log_off.py");
}

?>
```

```
import os
os.system("sudo /home/pi/datalogger/main")
```

Možnosť *Nastaviť RTC* spustí cez PHP skript jednoduchý program v jazyku C *clock.c*. V tomto programe je zavolaná funkcia *set_time_to_RTC*, ktorá nastaví hodiny v obvode hodín reálneho času zo systémových hodín Raspberry Pi.

```
#include "pcf8563.h"

int main (void){
    bcm2835_close();
    if(!bcm2835_init())
        printf("BCM2835_init failed!!!\n");

    set_time_to_RTC();
    bcm2835_close();
    return 0;
}
```

Posledné dve možnosti už nespúšťajú žiaden program, ale pracujú priamo s databázou. Cez obe tieto možnosti je možné sa cez PHP skript pripojiť do databázového systému a zadaným príkazom vykonať danú operáciu. V možnosti *Zmeniť periódu merania* zvolíme periódu, ktorú chceme nastaviť a potvrdíme. Zmenu periódy je vhodné meniť, keď je meranie zastavené. Následným spustením merania už bude datalogger zaznamenávať hodnoty s novou periódou.

```
<?php
if($stat == 'period'){
    $time = $_POST['time'];
    header("location:settings2.php");
    $servername = "localhost";
    $username = "datalogger";
    $password = "tomas6";
    $dbname = "temp";

    //Vytvorenie spojenia
    $conn = new mysqli($servername, $username, $password,
                      $dbname);

    //Overenie spojenia
    if ($conn->connect_error){
        die("Connection failed: ".$conn->connect_error);
    }
}
```

```
}  
$sql = "UPDATE period SET PERIOD='$time' WHERE ID='1'";  
$conn->query($sql);  
$conn->close();  
}  
?>
```

Možnosť *Vymazanie databázy* sa od predchádzajúcej možnosti líši len príkazmi pre MySQL. Príkaz *TRUNCATE* vymaže hodnoty v tabuľke tak, aby vymazalo aj index riadku a hlavička tabuľky sa nevymaže.

```
<?php  
$sql = "TRUNCATE TABLE digital_inputs";  
$conn->query($sql);  
$sql = "TRUNCATE TABLE analog_inputs";  
$conn->query($sql);?>
```

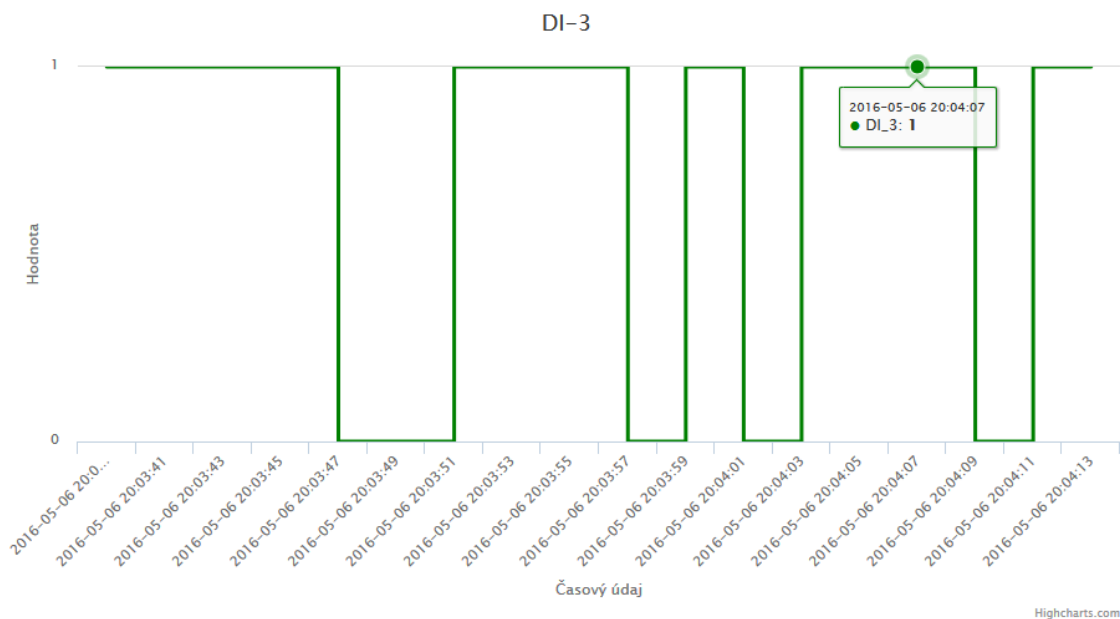
7 TEST FUNKČNOSTI

Správnú funkčnosť nami navrhnutého dataloggeru sme otestovali pripojením napätia na digitálny vstup *DI_3* a analógový vstup $0 - 10\text{ V}$. Ako zdroj napätia bol použitý regulovateľný zdroj napätia $0 - 30\text{ V} / 2\text{ A}$.

Na vstup *DI_3* sme náhodne pripájali napätie 24 V . Datalogger zaznamenával hodnoty s periódou merania 2 sekundy. Na nasledujúcich obrázkoch môžeme vidieť zaznamenané hodnoty v tabuľke, a hodnoty vykreslené do grafu. Vstupné napätie 24 V odpovedá hodnote 1. Graficky zobrazený nameraný bod si môžeme porovnať s hodnotou v tabuľke, v riadku s príslušným časovým údajom.

| Digitálny vstup DI-3 | | | |
|----------------------|------------|----------|---------|
| Tabuľka | | Graf | |
| ID | Dátum | Čas | Hodnota |
| 18 | 2016-05-06 | 20:04:13 | 1 |
| 17 | 2016-05-06 | 20:04:11 | 1 |
| 16 | 2016-05-06 | 20:04:09 | 0 |
| 15 | 2016-05-06 | 20:04:07 | 1 |
| 14 | 2016-05-06 | 20:04:05 | 1 |
| 13 | 2016-05-06 | 20:04:03 | 1 |
| 12 | 2016-05-06 | 20:04:01 | 0 |
| 11 | 2016-05-06 | 20:03:59 | 1 |
| 10 | 2016-05-06 | 20:03:57 | 0 |
| 9 | 2016-05-06 | 20:03:55 | 1 |
| 8 | 2016-05-06 | 20:03:53 | 1 |
| 7 | 2016-05-06 | 20:03:51 | 1 |
| 6 | 2016-05-06 | 20:03:49 | 0 |
| 5 | 2016-05-06 | 20:03:47 | 0 |

Obr. 7-1 Tabuľka pre vstup *DI_3*

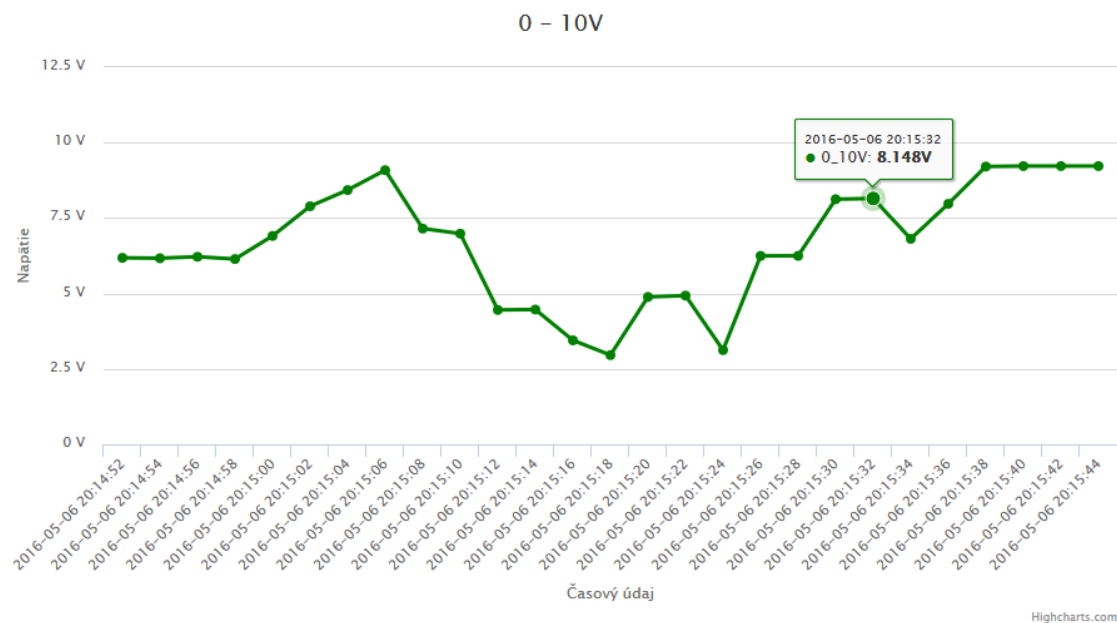


Obr. 7-2 Graf pre vstup DI_3

Na analógový vstup 0 – 10 V sme pripojili napätie v tomto rozsahu a náhodne ho menili. Perióda merania dataloggeru bola nastavená na hodnotu 1 sekunda. Tabuľka a grafické zobrazenie nameraných hodnôt sú na nasledujúcich obrázkoch.

| Analógový vstup 0-10 V | | | |
|------------------------|------------|----------|-------------|
| Tabuľka | | Graf | |
| ID | Dátum | Čas | Hodnota [V] |
| 27 | 2016-05-06 | 20:15:44 | 9.223 |
| 26 | 2016-05-06 | 20:15:42 | 9.223 |
| 25 | 2016-05-06 | 20:15:40 | 9.223 |
| 24 | 2016-05-06 | 20:15:38 | 9.207 |
| 23 | 2016-05-06 | 20:15:36 | 7.971 |
| 22 | 2016-05-06 | 20:15:34 | 6.811 |
| 21 | 2016-05-06 | 20:15:32 | 8.148 |
| 20 | 2016-05-06 | 20:15:30 | 8.124 |
| 19 | 2016-05-06 | 20:15:28 | 6.249 |
| 18 | 2016-05-06 | 20:15:26 | 6.245 |
| 17 | 2016-05-06 | 20:15:24 | 3.123 |
| 16 | 2016-05-06 | 20:15:22 | 4.929 |
| 15 | 2016-05-06 | 20:15:20 | 4.885 |
| 14 | 2016-05-06 | 20:15:18 | 2.958 |

Obr. 7-3 Tabuľka pre vstup 0 - 10 V



Obr. 7-4 Graf pre vstup 0 - 10 V

8 ZÁVER

Cieľom tejto diplomovej práce bolo navrhnuť Datalogger s Ethernet rozhraním. Ako riadiaci člen celého dataloggeru bol použitý Raspberry Pi 2 s procesorom ARM Cortex-A7 s frekvenciou 900 MHz od firmy Raspberry Foundation.

Datalogger je vybavený ôsmymi digitálnymi vstupmi pre napätie 24 V, dvoma napäťovými analógovými vstupmi z rozsahmi napätí 0 – 5 V a 0 – 10 V a jedným prúdovým analógovým vstupom 4 – 20 mA. Tieto vstupy sú upravené na napätie s vhodným rozsahom pre AD prevodník MCP3204.

Ďalej obsahuje 2 prepínače na zapnutie, LED diódy na signalizáciu stavov, hodiny reálneho času, stabilizátor napätia, galvanické oddelenia vstupov a záložný zdroj vo forme gombíkovej batérie. Datalogger používa komunikačné ethernetové rozhranie Raspberry Pi a taktiež pamäťovú kartu na uloženie nameraných hodnôt.

Datalogger je navrhnutý na dosku plošných spojov s rozmermi 57,77 x 66,35 mm v programe Eagle. Všetky súčiastky okrem batérie sú umiestnené vo vrchnej strane dosky. Osadzované boli ručne. Následne boli všetky komponenty oživené.

Raspberry Pi používa operačný systém Raspbian Wheezy. Datalogger bol programovaný vo viacerých jazykoch. Na obsluhu obvodu AD prevodníka pripojeného k riadiacej jednotke cez rozhranie SPI bola vytvorená knižnica datalogger a na obsluhu obvodu hodín reálneho času, ktoré sú pripojené cez zbernicu I²C, knižnica pcf8563. Obe knižnice sú napísané v jazyku C. Hlavný program je napísaný v jazyku C.

Namerané hodnoty sú zaznamenávané do databázového systému MySQL. Okrem hodnôt, ktoré datalogger zaznamená, je do databázy uložený aj časový údaj daného záznamu.

Jedným z bodov zadania bolo riadiť datalogger pomocou webového rozhrania. V Raspberry Pi je nainštalovaný webový server Apache, ktorý nám toto riadenie umožňuje. Na administráciu a riadenie dataloggeru je vytvorených niekoľko webových stránok v kombinácii značkovacieho jazyka HTML, skriptovacieho jazyka PHP a jazyka na tvorbu dynamických objektov JavaScript. Na webových stránkach sú namerané dáta načítané z databázy a zobrazené vo forme tabuľky alebo grafu. V práci tak boli splnené všetky body zadania.

Pri postupe práce sa vyskytol problém s obvodom hodín reálneho času PCF8563. Obvod zapojený podľa dokumentácie mal chybu presnosti hodín približne pätnásť sekúnd na deň. Problém sme riešili paralelným pripojením kondenzátora s kapacitou 18 pF ku kryštálu. Hodnotu kapacity kondenzátora sme určili pomocou osciloskopu, kedy frekvencia 1 Hz bola najpresnejšia. Presnosť obvodu sa zvýšila na jednu sekundu na týždeň.

Presnosť merania analógových vstupov bola porovnávaná s digitálnym multimetrom Fluke 1587. Veľkosť odchýlky bola 3 mV.

Významnou sa táto práca stáva nie len z pohľadu osobného rastu a zdokonalenia sa vo vytváraní webových stránok a programovacích jazykov, ale predovšetkým sa ukázalo, že použitý riadiaci člen Raspberry Pi je vhodný komponent pre datalogger.

Navrhnutý datalogger by sa mohol používať pri diagnostikách problémov v priemysle. Vďaka rýchlemu vývoju vedy a techniky sa stala bežnou súčasťou priemyselných strojov internetová sieť a ethernetové rozhranie, čo prináša obrovské možnosti a výhody riadenia a kontroly strojov neobmedzene na čas a vzdialenosť.

LITERATÚRA

- [1] Datalogger. In: Wikipedia: The Free Encyclopedia [online]. 2001-, 19.12.2014 [cit. 2016-05-16]. Dostupné z: http://en.wikipedia.org/wiki/Data_logger
- [2] VÁGNER, Martin. Malý datalogger s flash kartou [online]. Brno, 2008 [cit. 2016-05-16]. Dostupné z: http://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=7309. Bakalárska práca. VUT Brno.
- [3] ČEJKA M.: Měření v elektrotechnice. VUT, Brno 2002
- [4] Miroslav Novák .com. DataLogger [online]. 2012 [cit. 2016-05-16]. Dostupné z: http://www.miroslavnovak.com/datalogger_cz.php
- [5] Raspberry Pi. *Help* [online]. [cit. 2016-05-07]. Dostupné z: <https://www.raspberrypi.org/help/>
- [6] ADUM 4150. *5 kV, 6 -Channel, SPI Isolator Digital Isolator for SPI with Delay Clock* [online]. 2015 [cit. 2016-05-07]. Dostupné z: <http://www.analog.com/media/en/technical-documentation/data-sheets/ADuM4150.pdf>
- [7] PC World. *Raspberry Pi 2 review: The revolutionary \$35 micro-PC, supercharged* [online]. 2016 [cit. 2016-05-07]. Dostupné z: <http://www.pcworld.com/article/2886260/raspberry-pi-2-review-the-revolutionary-35-micro-pc-supercharged.html>
- [8] Bcm2835. *C library for Broadcom BCM 2835 as used in Raspberry Pi* [online]. 2016 [cit. 2016-05-07]. Dostupné z: <http://www.airspayce.com/mikem/bcm2835/index.html>
- [9] TLP283, TLP283-4: Datasheet [online]. 2007 [cit. 2015-01-06]. Dostupné z: <http://www.mouser.com/ds/2/408/4197-54115.pdf>
- [10] TLV2764: Datasheet [online]. 2000 [cit. 2015-01-06]. Dostupné z: <http://www.ti.com/lit/ds/symlink/tlv2764.pdf>
- [11] MCP 3204/3208: *Datasheet* [online]. 2008 [cit. 2016-05-07]. Dostupné z: <http://www.airspayce.com/mikem/bcm2835/index.html>

- [12] *REF2940: Datasheet* [online]. 2014 [cit. 2016-05-07]. Dostupné z: <http://www.ti.com/lit/ds/symmlink/ref2940.pdf>
- [13] *PCF8563: Datasheet* [online]. 2015 [cit. 2016-05-07]. Dostupné z: <http://www.ti.com/lit/ds/symmlink/ref2940.pdf>
- [14] *Raspberry Pi: Downloads* [online]. 2015 [cit. 2016-05-07]. Dostupné z: <http://www.raspberrypi.org/downloads/>
- [15] *W3SCHOOL: HTML Tutorial* [online]. 2016 [cit. 2016-05-07]. Dostupné z: <http://www.w3schools.com/>
- [16] *LP3985 Micropower: Datasheet* [online]. 2016 [cit. 2016-05-07]. Dostupné z: <http://www.ti.com/lit/ds/symmlink/lp3985.pdf>
- [17] *ADUM 5000: Datasheet* [online]. 2012 [cit. 2016-05-07]. Dostupné z: <http://www.analog.com/media/en/technical-documentation/data-sheets/ADuM5000.pdf>
- [18] WILLIAMS, Hugh E. a David LANE. *Programujeme webové aplikace pomocí PHP a MySQL*. Praha: Computer Press, 2002. Podrobný průvodce tvůrce WWW stránek. ISBN 80-722-6760-4.
- [19] *HIGHSTOCK: Download* [online]. 2016 [cit. 2016-05-07]. Dostupné z: <http://www.highcharts.com/products/highstock>
- [20] HARMS, Daryl D. a Kenneth MCDONALD. *Začínáme programovat v jazyce Python*. 2., opr. vyd. Překlad Ivo Fořt, Lubomír Škapa. Brno: Computer Press, 2008. Podrobný průvodce tvůrce WWW stránek. ISBN 978-80-251-2161-0.

ZOZNAM SKRATIEK A SYMBOLOV

| | |
|------------------|-------------------------------------|
| SD | Secure Digital |
| USB | Universal Serial Bus |
| PC | Personal Computer |
| CAN | Controller Area Network |
| A/D | Analógovo digitálny |
| RTC | Real Time Clock |
| CPU | Central Processnig Unit |
| RAM | Random Access Memory |
| SPI | Serial Peripheral Interface |
| MOSI | Master Output Slave Input |
| MISO | Master Input Slave Output |
| SCLK | Clock |
| GPIO | General Purpose Input Output |
| SDA | Serial Data |
| SCL | Clock |
| CS | Chip Select |
| SS | Slave Select |
| INT | Interrupt |
| MMC | Multi Media Card |
| GPRS | General Packet Radio Service |
| I ² C | Inter-Integrated Circuit |
| PHP | HyperText Preprocessor |
| IP | Internet Protocol |
| MSB | Most Significant Bit |
| LSB | Least Significant Bit |
| HTML | HyperText Marker Language |
| DHCP | Dynamic Host Configuration Protocol |

ZOZNAM PRÍLOH

A Zoznam súčiastok

B Schéma zapojenia

C Návrh plošného spoja TOP, BOTTOM

D Osadzovací plán súčiastok

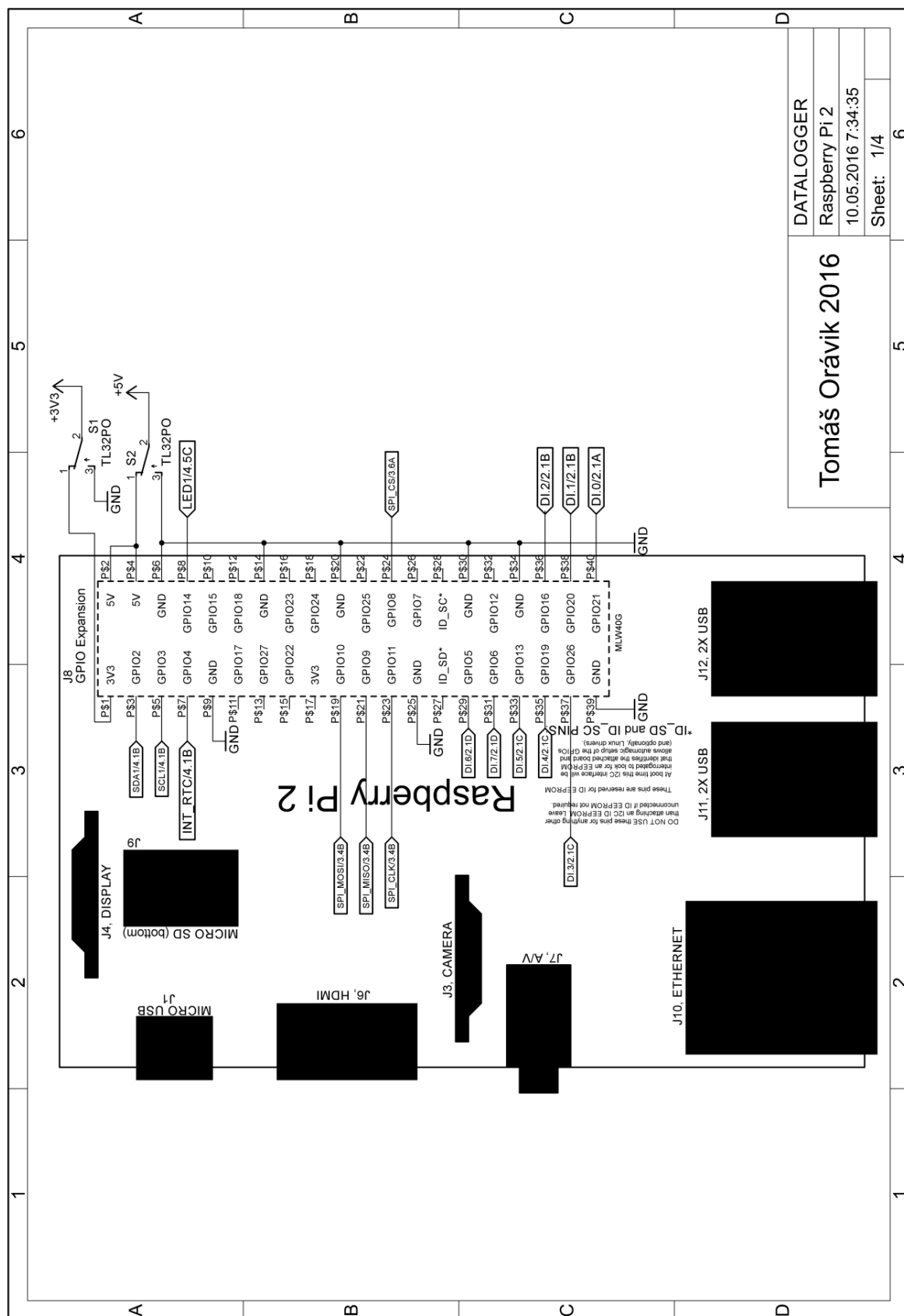
E Fotodokumentácia

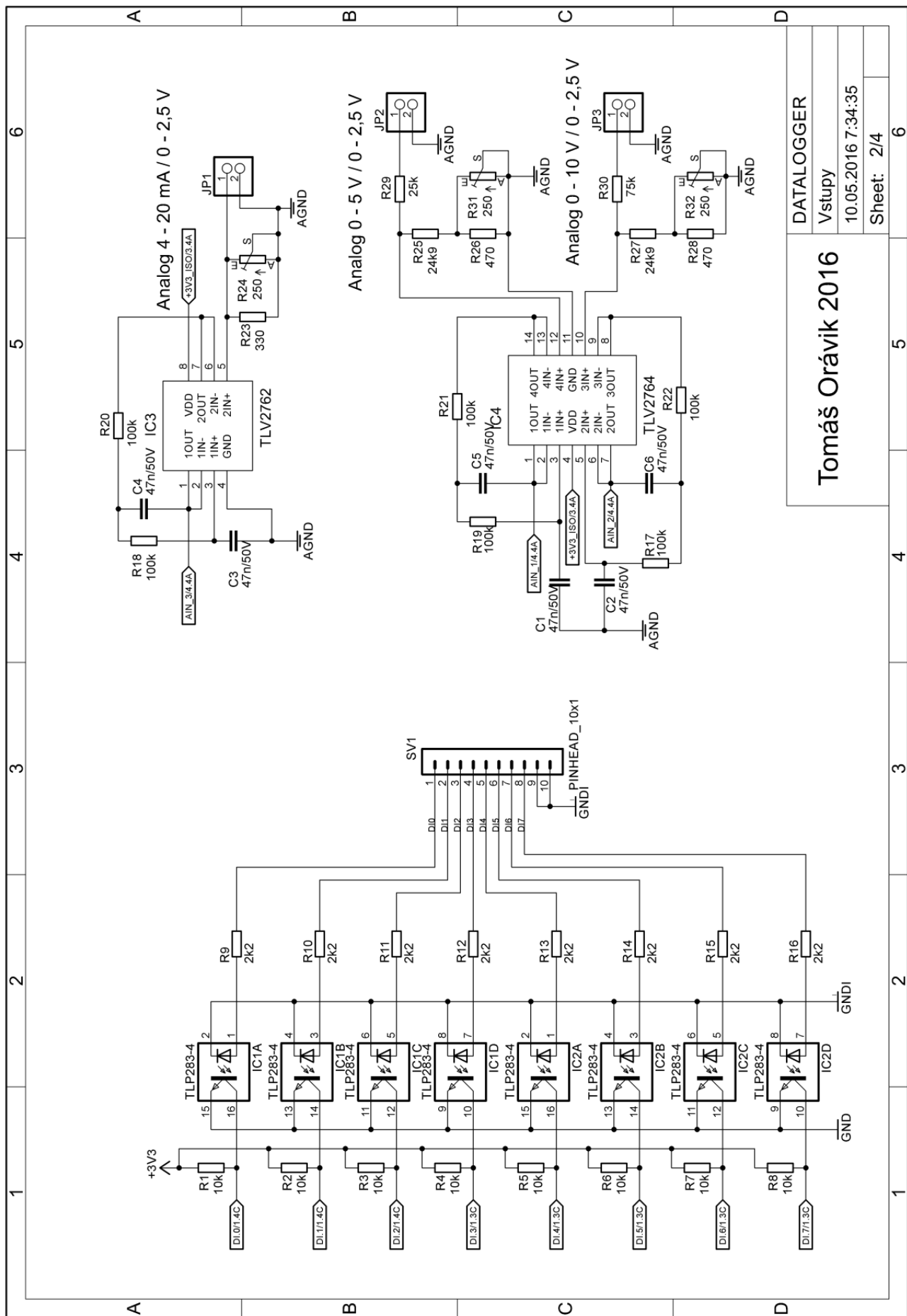
F CD

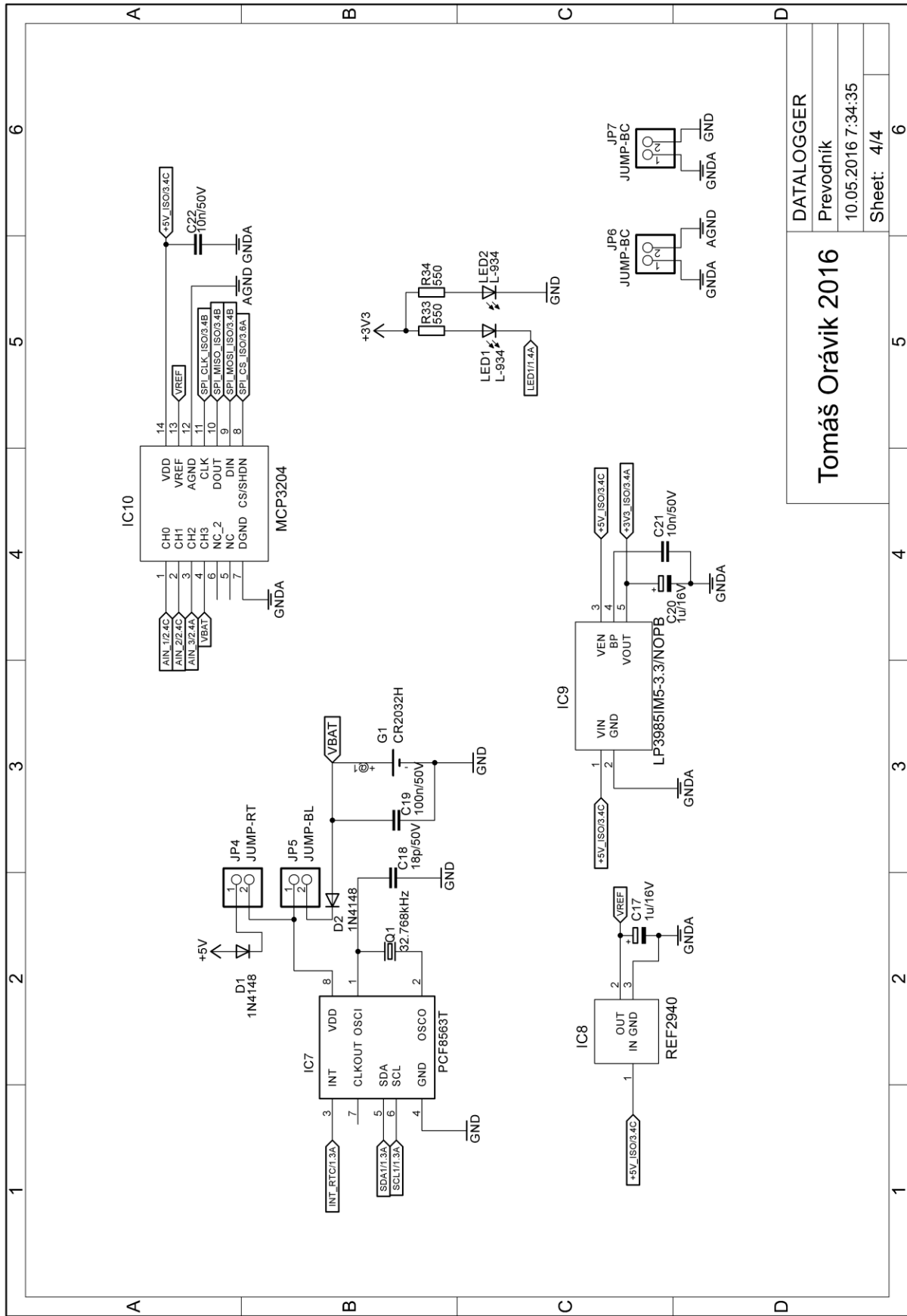
A. Zoznam súčiastok

| Súčiastka | Počet | Hodnota | Puzdro |
|---------------------------------------|-------|--------------|--------------|
| R17, R18, R19, R20, R21, R22 | 6 | 100k | R0805 |
| C7, C10, C11, C12, C14, C15, C19 | 7 | 100n/50V | C0805 |
| R1, R2, R3, R4, R5, R6, R7, R8 | 8 | 10k | R0805 |
| C21, C22 | 2 | 10n/50V | C0805 |
| C8, C9, C13, C16 | 4 | 10u/10V | C0805 |
| C18 | 1 | 18p/50V | C0805 |
| D1, D2 | 2 | 1N4148 | SOD80C |
| C17, C20 | 2 | 1u/16V | C025-025X050 |
| R25, R27 | 2 | 24k9 | R0805 |
| R24, R31, R32 | 3 | 250 | CA6V |
| R29 | 1 | 25k | R0805 |
| R9, R10, R11, R12, R13, R14, R15, R16 | 8 | 2k2 | R0805 |
| Q1 | 1 | 32.768kHz | TC26V |
| R23 | 1 | 330 | R0805 |
| R26, R28 | 2 | 470 | R0805 |
| C1, C2, C3, C4, C5, C6 | 6 | 47n/50V | C0805 |
| R33, R34 | 2 | 550 | R0805 |
| R30 | 1 | 75k | R0805 |
| IC6 | 1 | ADUM4150 | RI_20_1 |
| IC5 | 1 | ADUM5000 | RW_16 |
| G1 | 1 | CR2032H | CR2032H |
| JP6, JP7 | 2 | JUMP-BC | 1X02 |
| JP5 | 1 | JUMP-BL | 1X02 |
| JP4 | 1 | JUMP-RT | 1X02 |
| LED1, LED2 | 2 | L-934 | LED3MM |
| IC9 | 1 | LP3985IM5 | MF05A |
| IC10 | 1 | MCP3204 | SOIC14_N_MC |
| IC7 | 1 | PCF8563T | SO8 |
| SV1 | 1 | PINHEAD_10x1 | MA10-1 |
| IC8 | 1 | REF2940 | DBZ3 |
| S1, S2 | 2 | TL32PO | TL3XPO |
| IC1, IC2 | 2 | TLP283-4 | MINI-FLAT-16 |
| IC3 | 1 | TLV2762 | D8 |
| IC4 | 1 | TLV2764 | D14 |
| JP1, JP2, JP3 | 3 | | 1X02 |
| PCB1 | 1 | RPI2 | RPI2 |

B. Schéma zapojenia

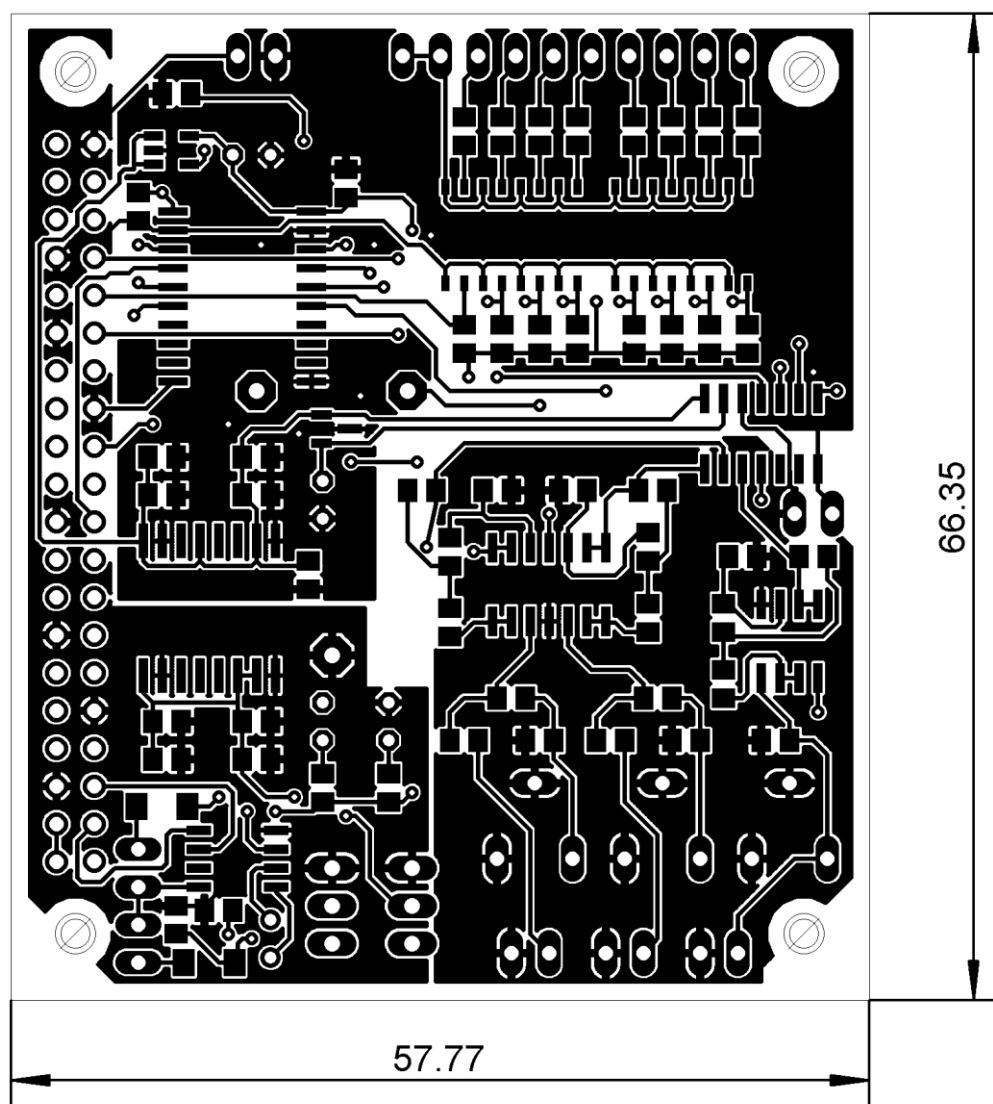


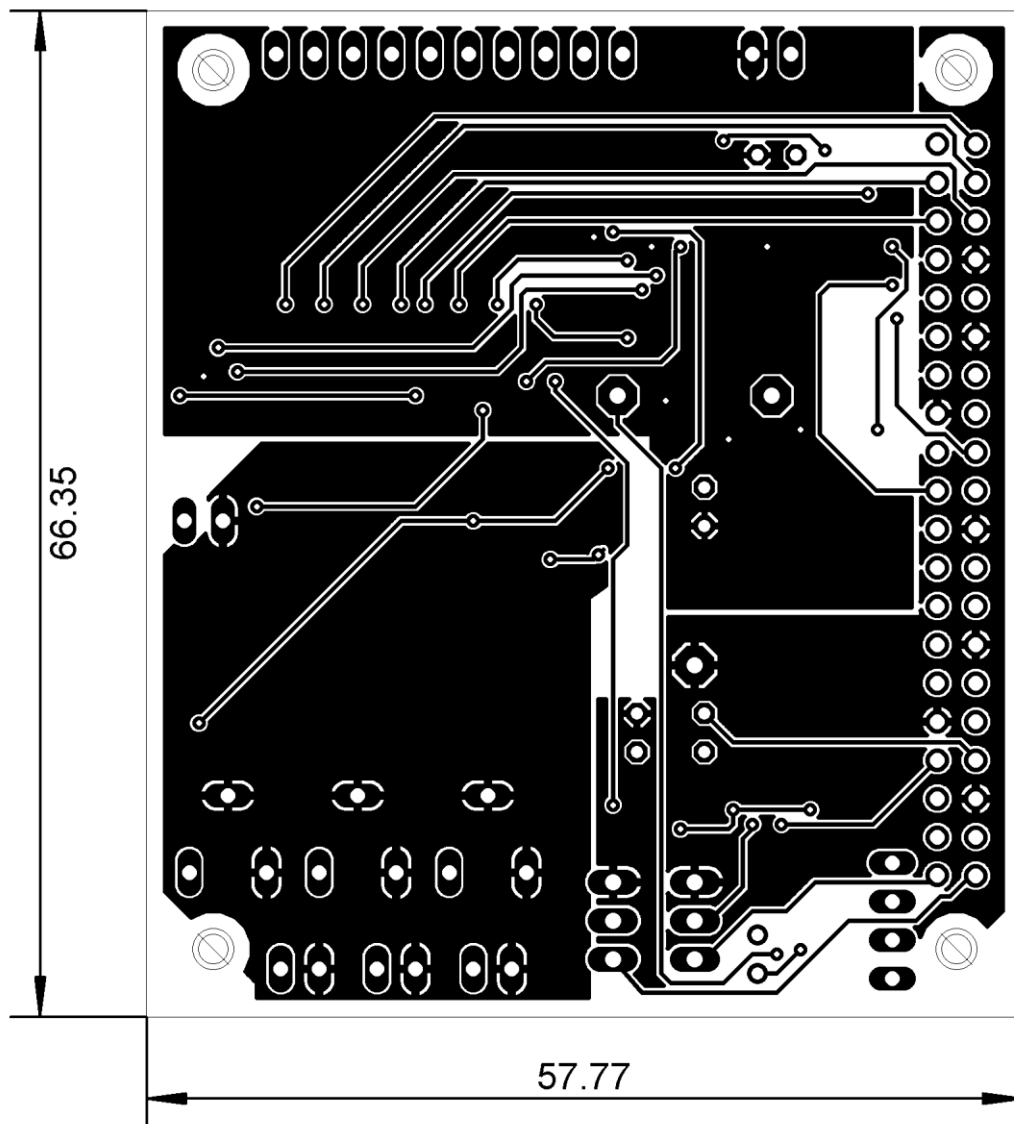




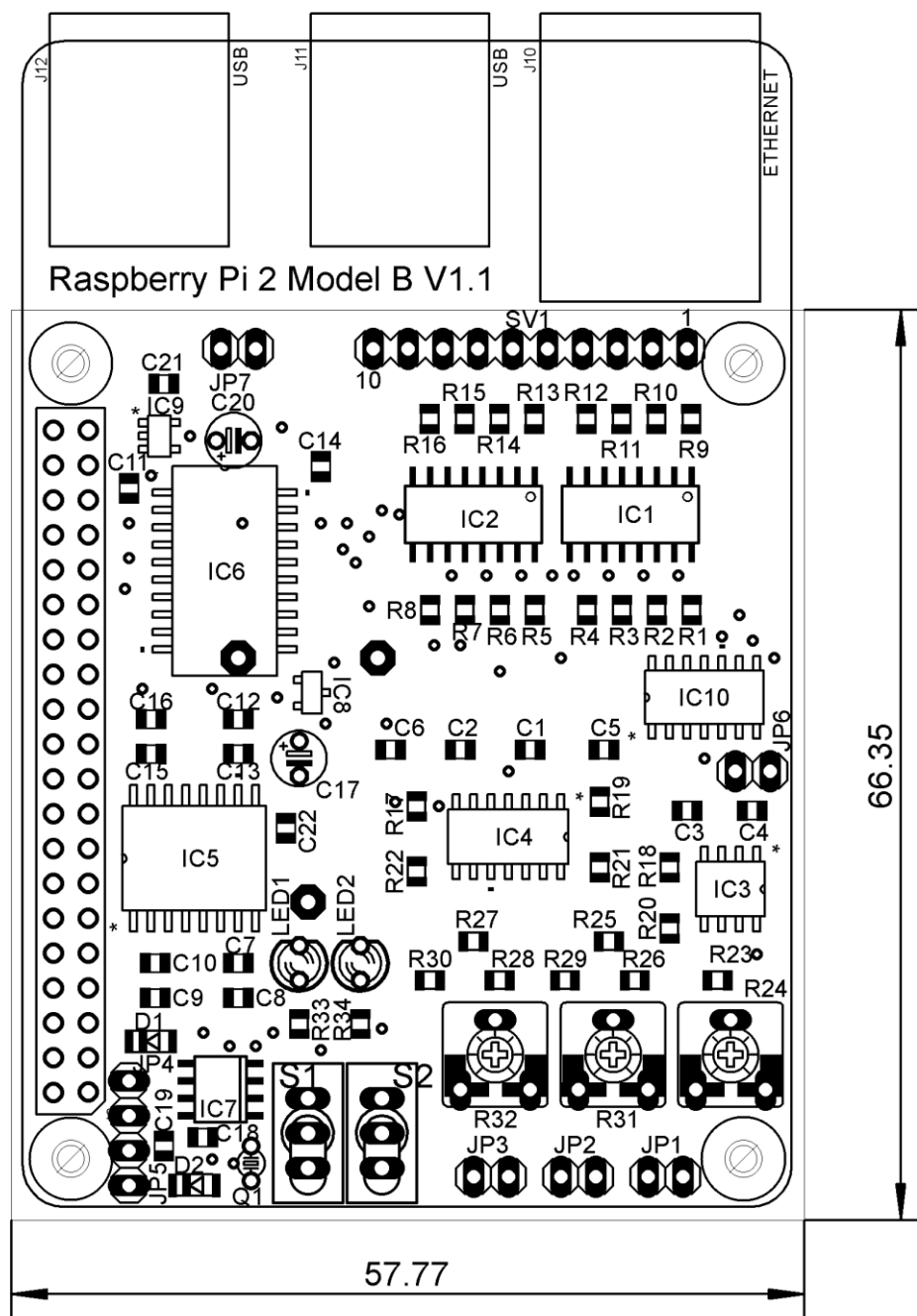
| | |
|--------------------|---|
| Tomáš Orávík 2016 | |
| DATALOGGER | |
| Prevodník | |
| 10.05.2016 7:34:35 | |
| Sheet: 4/4 | 6 |

C. Návrh plošného spoja TOP, BOTTOM





D. Osadzovací plán súčiastok



E. Fotodokumentácia

